



Open Source Used In Debug-container elpaso

Cisco Systems, Inc.

www.cisco.com

Cisco has more than 200 offices worldwide.
Addresses, phone numbers, and fax numbers
are listed on the Cisco website at
www.cisco.com/go/offices.

Text Part Number: 78EE117C99-2812803066

This document contains licenses and notices for open source software used in this product. With respect to the free/open source software listed in this document, if you have any questions or wish to receive a copy of any source code to which you may be entitled under the applicable free/open source license(s) (such as the GNU Lesser/General Public License), please submit this [form](#).

In your requests please include the following reference number 78EE117C99-2812803066

Contents

1.1 simple launcher 1.1.0.14

1.1.1 Available under license

1.2 ipset 7.24-r0

1.2.1 Available under license

1.3 bridgeutils 1.7.1-r2

1.3.1 Available under license

1.4 typeguard 4.3.0

1.4.1 Available under license

1.5 util-linux 2.41-r9

1.5.1 Available under license

1.6 libxml2 2.13.9-r0

1.6.1 Available under license

1.7 idna 15.1.0

1.7.1 Available under license

1.8 libbsd 0.12.2-r0

1.8.1 Available under license

1.9 python-requests 2.32.3

1.9.1 Available under license

1.10 packaging 25.0

1.10.1 Available under license

1.11 pax-utils 1.3.8-r1

1.11.1 Available under license

1.12 elfutils 0.193-r0

1.12.1 Available under license

1.13 valgrind 3.25.1-r1

1.13.1 Available under license

1.14 expat 2.7.3-r0

1.14.1 Available under license

1.15 more-itertools 10.3.0

1.15.1 Available under license

1.16 krb5-conf 1.0-r2

1.16.1 Available under license

1.17 libffi 3.4.8-r0

1.17.1 Available under license

1.18 py3-pip 25.1.1-r0

1.18.1 Available under license

1.19 libpsl 0.21.5-r3

1.19.1 Available under license

1.20 mpdecimal 4.0.1-r0

1.20.1 Available under license

1.21 jaraco-functools 4.0.1

1.21.1 Available under license

1.22 heimdal 18.0.2

1.22.1 Available under license

1.23 apr-util 1.6.3-r1

1.23.1 Available under license

1.24 zlib 1.3.1-r2

1.24.1 Available under license

1.25 gmp 6.3.0-r3

1.25.1 Available under license

1.26 libcap-ng 0.8.5-r0

1.26.1 Available under license

1.27 libnetfilter-queue 1.0.5-r2

1.27.1 Available under license

1.28 speexdsp 1.2.1-r2

1.28.1 Available under license

1.29 sqlite 3.49.2-r1

1.29.1 Available under license

1.30 libmnl 1.0.5-r2

1.30.1 Available under license

1.31 nftables 1.1.3-r0

1.31.1 Available under license

1.32 libgcrypt 1.10.3-r1

1.32.1 Available under license

1.33 git 2.49.1-r0

1.33.1 Available under license

1.34 net-snmp 5.9.4-r1

1.34.1 Available under license

1.35 htop 3.4.1-r0

1.35.1 Available under license

1.36 alpine-base 3.22.2-r0

1.36.1 Available under license

1.37 zipp 3.19.2

1.37.1 Available under license

1.38 futures 3.12

1.38.1 Available under license

1.39 libpkit 0.25.5-r2

1.39.1 Available under license

1.40 popt 1.19-r4

1.40.1 Available under license

1.41 glib 2.84.4-r0

1.41.1 Available under license

1.42 libnetfilter-contrack 1.4.8

1.42.1 Available under license

1.43 apr 1.7.5-r0

1.43.1 Available under license

1.44 pcre2 10.46-r0

1.44.1 Available under license

1.45 alpine-keys 2.5-r0

1.45.1 Available under license

1.46 distro 1.9.0

1.46.1 Available under license

1.47 tcptraceroute 1.5_beta7-r4

1.47.1 Available under license

1.48 libevent 2.1.12-r8

1.48.1 Available under license

1.49 wireshark 4.4.7-r0

1.49.1 Available under license

1.50 sysstat 12.7.6-r1

1.50.1 Available under license

1.51 libpcap 1.10.5-r1

1.51.1 Available under license

1.52 jaraco.collections 5.1.0

1.52.1 Available under license

1.53 libbpf 1.5.1-r0

1.53.1 Available under license

1.54 libgpg-error 1.55-r0

1.54.1 Available under license

1.55 ca-certificates 20250911-r0

1.55.1 Available under license

1.56 libidn 2.3.7-r0

1.56.1 Available under license

1.57 wheel 0.45.1

1.57.1 Available under license

1.58 lz4 1.10.0-r0

1.58.1 Available under license

1.59 mtr 0.96-r0

1.59.1 Available under license

1.60 iptables 1.8.11-r1

1.60.1 Available under license

1.61 ipvsadm 1.31-r3

1.61.1 Available under license

1.62 fping 5.3-r0

1.62.1 Available under license

1.63 bash 5.2.37-r0

1.63.1 Available under license

1.64 ethtool 6.14.1-r0

1.64.1 Available under license

1.65 my-test-package 1.0

1.65.1 Available under license

1.66 libnet 1.3-r0

1.66.1 Available under license

1.67 iputils 20240905-r0

1.67.1 Available under license

1.68 iptraf-ng 1.2.2-r0

1.68.1 Available under license

1.69 lua 5.4.7-r0

1.69.1 Available under license

1.70 curl 8.14.1-r2

1.70.1 Available under license

1.71 cpp 14.2.0-r6

1.71.1 Available under license

1.72 libnetfilter-cttimeout 1.0.1-r2

- 1.72.1 Available under license
- 1.73 bzip2 1.0.8-r6**
 - 1.73.1 Available under license
- 1.74 pyproject-hooks 1.2.0**
 - 1.74.1 Available under license
- 1.75 libnetfilter-cthelper 1.0.1-r2**
 - 1.75.1 Available under license
- 1.76 python-pip 25.1.1**
 - 1.76.1 Available under license
- 1.77 zsh 5.9-r5**
 - 1.77.1 Available under license
- 1.78 gnutls 3.8.8-r0**
 - 1.78.1 Available under license
- 1.79 musl 1.2.5-r10**
 - 1.79.1 Available under license
- 1.80 urllib3 1.26.20**
 - 1.80.1 Available under license
- 1.81 speedtest-cli 2.1.3-r7**
 - 1.81.1 Available under license
- 1.82 py3-packaging 25.0-r0**
 - 1.82.1 Available under license
- 1.83 python-certifi 2025.01.31**
 - 1.83.1 Available under license
- 1.84 iproute 6.15.0-r0**
 - 1.84.1 Available under license
- 1.85 libnftnl 1.2.9-r0**
 - 1.85.1 Available under license
- 1.86 e2fsprogs 1.47.2-r2**
 - 1.86.1 Available under license
- 1.87 libnfnetwork 1.0.2-r3**
 - 1.87.1 Available under license
- 1.88 platformdirs 4.2.2**
 - 1.88.1 Available under license
- 1.89 libuv 1.51.0-r0**
 - 1.89.1 Available under license
- 1.90 kerberos 1.21.3-r0**
 - 1.90.1 Available under license
- 1.91 importlib-metadata 8.0.0**
 - 1.91.1 Available under license

1.92 libmaxminddb 1.9.1-r0

1.92.1 Available under license

1.93 iperf 2.2.0-r0

1.93.1 Notifications

1.93.2 Available under license

1.94 nghttp2 1.65.0-r0

1.94.1 Available under license

1.95 apk-tools 2.14.9-r3

1.95.1 Available under license

1.96 netcat-openbsd 1.229.1-r0

1.96.1 Available under license

1.97 fdisk 2.41.0

1.97.1 Available under license

1.98 libtasn 4.20.0-r0

1.98.1 Available under license

1.99 musl-fts 1.2.7-r6

1.99.1 Available under license

1.100 rich 13.9.4

1.100.1 Available under license

1.101 backports.tarfile 1.2.0

1.101.1 Available under license

1.102 platformdirs 4.3.7

1.102.1 Available under license

1.103 cachecontrol 0.14.2

1.103.1 Available under license

1.104 cachecontrol 0.14.1

1.104.1 Available under license

1.105 tre 1.2.5-r8

1.105.1 Available under license

1.106 python-pip 25.0.1

1.106.1 Available under license

1.107 jansson 2.14.1-r0

1.107.1 Available under license

1.108 py3-setuptools 80.9.0-r0

1.108.1 Available under license

1.109 libcap 2.76-r0

1.109.1 Available under license

1.110 openssl 3.5.4-r0

1.110.1 Available under license

1.111 pyparsing 3.2.3

1.111.1 Available under license

1.112 ltrace 0.7.3-r6

1.112.1 Available under license

1.113 libnetfilter-contrack 1.1.0-r0

1.113.1 Available under license

1.114 truststore 0.10.0

1.114.1 Available under license

1.115 json-c 0.18-r1

1.115.1 Available under license

1.116 gettext 0.24.1-r0

1.116.1 Available under license

1.117 liburcu 0.15.2-r0

1.117.1 Available under license

1.118 libunistring 1.3-r0

1.118.1 Available under license

1.119 gobject-introspection 1.84.0-r0

1.119.1 Available under license

1.120 judy 1.0.5-r1

1.120.1 Available under license

1.121 libnl3 3.11.0-r0

1.121.1 Available under license

1.122 zstd 1.5.7-r0

1.122.1 Available under license

1.123 vim 9.1.1566-r0

1.123.1 Available under license

1.124 protobuf-c 1.5.2-r0

1.124.1 Available under license

1.125 busybox 1.37.0-r19

1.125.1 Available under license

1.126 ngrep 1.47-r4

1.126.1 Available under license

1.127 xz 5.8.1-r0

1.127.1 Available under license

1.128 py3-parsing 3.2.3-r0

1.128.1 Available under license

1.129 lua5.4 5.4.7-r0

1.129.1 Available under license

1.130 utmps 0.1.3.1-r0

1.130.1 Available under license
1.131 autocommand 2.2.2
1.131.1 Available under license
1.132 jaraco.text 3.12.1
1.132.1 Available under license
1.133 python-certifi 2024.08.30
1.133.1 Available under license
1.134 dhcping 1.2-r3
1.134.1 Available under license
1.135 readline 8.2.13-r1
1.135.1 Available under license
1.136 tomli 2.0.1
1.136.1 Available under license
1.137 liboping 1.10.0-r2
1.137.1 Available under license
1.138 libmd 1.1.0-r0
1.138.1 Available under license
1.139 iftop 0.17-r11
1.139.1 Available under license
1.140 python-setuptools 80.9.0
1.140.1 Available under license
1.141 tomli 2.2.1
1.141.1 Available under license
1.142 oniguruma 6.9.10-r0
1.142.1 Available under license
1.143 arptables 1.8.11
1.143.1 Available under license
1.144 libedit 20250104.3.1-r1
1.144.1 Available under license
1.145 gdbm 1.24-r0
1.145.1 Available under license
1.146 skalibs 2.14.4.0-r0
1.146.1 Available under license
1.147 wget 1.25.0-r1
1.147.1 Available under license
1.148 libeconf 0.6.3-r0
1.148.1 Available under license
1.149 tmux 3.5a-r1
1.149.1 Available under license

1.150 fstrm 0.6.1-r4

1.150.1 Available under license

1.151 keyutils 1.6.3-r4

1.151.1 Available under license

1.152 openssh 10.0_p1-r9

1.152.1 Notifications

1.152.2 Available under license

1.153 scapy 2.6.1-r0

1.153.1 Available under license

1.154 file 5.46-r2

1.154.1 Available under license

1.155 libverto 0.3.2-r2

1.155.1 Available under license

1.156 nftables-vim 0_git20200629-r1

1.156.1 Available under license

1.157 alpine-baselayout 3.7.0-r0

1.157.1 Available under license

1.158 conntrack-tools 1.4.8-r0

1.158.1 Available under license

1.159 brotli 1.1.0-r2

1.159.1 Available under license

1.160 llvm 18.1.8-r5

1.160.1 Available under license

1.161 libssh 0.11.2-r0

1.161.1 Available under license

1.162 tcpdump 4.99.5-r1

1.162.1 Available under license

1.163 python 3.12.12-r0

1.163.1 Available under license

1.164 jaraco.context 5.3.0

1.164.1 Available under license

1.165 libsmi 0.5.0-r4

1.165.1 Available under license

1.166 strace 6.13-r0

1.166.1 Available under license

1.167 libnettle6 3.10.1-r0

1.167.1 Available under license

1.168 distlib 0.3.9

1.168.1 Available under license

1.169 clang 18.1.8-r5

1.169.1 Available under license

1.170 inflect 7.3.1

1.170.1 Available under license

1.171 msgpack 1.1.0

1.171.1 Available under license

1.172 zmap 4.2.0-r0

1.172.1 Available under license

1.173 bcc 0.34.0-r0

1.173.1 Available under license

1.174 freeipmi 18.0.2

1.174.1 Available under license

1.175 ldns 1.8.4-r0

1.175.1 Available under license

1.176 typing-extensions 4.12.2

1.176.1 Available under license

1.177 pam 1.7.0-r4

1.177.1 Available under license

1.178 packaging 24.2

1.178.1 Available under license

1.179 ncurses 6.5_p20250503-r0

1.179.1 Available under license

1.180 truststore 0.10.1

1.180.1 Available under license

1.181 c-ares 1.34.5-r0

1.181.1 Available under license

1.182 socat 1.8.0.3-r1

1.182.1 Available under license

1.183 clang18 18.1.8-r5

1.183.1 Available under license

1.184 bird 3.1.4-r0

1.184.1 Available under license

1.185 llvm18 18.1.8-r5

1.185.1 Available under license

1.186 jq 1.8.1-r0

1.186.1 Available under license

1.187 busybox 1.37.0-r20

1.187.1 Available under license

1.188 bind 9.20.16-r0

1.1 simple launcher 1.1.0.14

1.1.1 Available under license :

A. HISTORY OF THE SOFTWARE

=====

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see <http://www.cwi.nl>) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see <http://www.cnri.reston.va.us>) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations (now Zope Corporation, see <http://www.zope.com>). In 2001, the Python Software Foundation (PSF, see <http://www.python.org/psf/>) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation is a sponsoring member of the PSF.

All Python releases are Open Source (see <http://www.opensource.org> for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

Release	Derived from	Year	Owner	GPL-compatible? (1)
0.9.0 thru 1.2		1991-1995	CWI	yes
1.3 thru 1.5.2	1.2	1995-1999	CNRI	yes
1.6	1.5.2	2000	CNRI	no
2.0	1.6	2000	BeOpen.com	no
1.6.1	1.6	2001	CNRI	yes (2)
2.1	2.0+1.6.1	2001	PSF	no
2.0.1	2.0+1.6.1	2001	PSF	yes
2.1.1	2.1+2.0.1	2001	PSF	yes
2.2	2.1.1	2001	PSF	yes
2.1.2	2.1.1	2002	PSF	yes
2.1.3				

2.1.2	2002	PSF	yes	
2.2.1	2.2	2002	PSF	yes
2.2.2	2.2.1	2002	PSF	yes
2.2.3	2.2.2	2003	PSF	yes
2.3	2.2.2	2002-2003	PSF	yes
2.3.1	2.3	2002-2003	PSF	yes
2.3.2	2.3.1	2002-2003	PSF	yes
2.3.3	2.3.2	2002-2003	PSF	yes
2.3.4	2.3.3	2004	PSF	yes
2.3.5	2.3.4	2005	PSF	yes
2.4	2.3	2004	PSF	yes
2.4.1	2.4	2005	PSF	yes
2.4.2	2.4.1	2005	PSF	yes
2.4.3	2.4.2	2006	PSF	yes
2.4.4	2.4.3	2006	PSF	yes
2.5	2.4	2006	PSF	yes
2.5.1	2.5	2007	PSF	yes
2.5.2	2.5.1	2008	PSF	yes
2.5.3	2.5.2	2008	PSF	yes
2.6	2.5	2008	PSF	yes
2.6.1	2.6	2008	PSF	yes
2.6.2	2.6.1	2009	PSF	yes
2.6.3	2.6.2	2009	PSF	yes
2.6.4	2.6.3	2009	PSF	yes
2.6.5	2.6.4	2010	PSF	yes
3.0	2.6	2008	PSF	yes
3.0.1	3.0	2009	PSF	yes
3.1	3.0.1	2009	PSF	yes
3.1.1	3.1	2009	PSF	yes
3.1.2	3.1	2010	PSF	yes
3.2	3.1	2010	PSF	yes

Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

=====

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.
2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010 Python Software Foundation; All Rights Reserved" are retained in Python alone or in any derivative version prepared by Licensee.
3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.
4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.
5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.
6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.
7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF

trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3.
BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of

agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at <http://www.pythonlabs.com/logos.html> may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the Internet using the following URL: <http://hdl.handle.net/1895.22/1013>".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS"

basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6.

This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted,

provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

UNRECOGNIZED LICENSE; MD5 sum: 82b32a7a0e773c8801056f80aaae2a44

UNRECOGNIZED LICENSE; MD5 sum: 25e3201901e018ec6d4724df0a74d278

=====

Distutils2 Contributors

=====

The Distutils2 project was started by Tarek Ziad and is currently maintained by ric Araujo. Many people have contributed to the project.

distlib has started off using some of the code from distutil2.

If you're making a patch, please add your name below in alphabetical order, and welcome into the Fellowship of the Packaging!

Thanks to:

- Rajiv Abraham
- Ali Afshar
- David Barnett
- Pior Bastida
- Anthony Baxter
- Erik Bray
- C. Titus Brown
- Francisco Martn Brugu
- Nicolas Cadou
- Godefroid Chapelle
- Julien Courteau
- Christophe Combelles
- Jason R. Coombs
- Pierre-Yves David
- Ned Deily
- Konrad DeLong
- Josip Djolonga

- John Edmonds
- Andr Espaze
- Boris Feld
- Andrew Francis
- Hallvard B Furuseth
- Patrice Gauthier
- Yannick Gingras
- Filip Gruszczyski
- Walker Hale IV
- Alexandre Hamelin
- Kelsey Hightower
- Thomas Holmes
- Preston Holmes
- Christian Hudon
- Julien Jehannet
- Jeremy Kloth
- Thomas Kluyver
-
- Amos Latteier
- Mathieu Leduc-Hamel
- Pierre Paul Lefebvre
- Tshepang Lekhonkhobe
- Alain Leufroy
- Janusz Lewandowski
- Martin von Lwis
- Hugo Lopes Tavares
- Guillermo Lpez-Anglada
- Justin Love
- Simon Mathieu
- Carl Meyer
- Alexis Mtaireau
- Julien Miotte
- Zubin Mithra
- Derek McTavish Mounce
- Paul Moore
- Michael Mulich
- Louis Munro
- Gal Pasgrimaud
- George Peristerakis
- Mathieu Perreault
- Guillaume Pratte
- Sean Reifschneider
- Antoine Reversat
- Arc Riley
- C. Anthony Risinger
- Elson Rodriguez
- Luis Rojas
- Erik Rose

- Brian Rosner
- Vinay Sajip
- Victor Stinner
- Alexandre Vassalotti
- Nadeem Vawda

Metadata-Version: 1.0

Name: setuptools

Version: 0.6c9

Summary: Download, build, install, upgrade, and uninstall Python packages -- easily!

Home-page: <http://pypi.org/pypi/setuptools>

Author: Phillip J. Eby

Author-email: distutils-sig@python.org

License: PSF or ZPL

Description: =====

Installing and Using Setuptools

=====

.. contents:: **Table of Contents**

Installation Instructions

Windows
=====

Install setuptools using the provided ``.exe`` installer. If you've previously installed older versions of setuptools, please delete all ``setuptools\*.egg`` and ``setuptools.pth`` files from your system's ``site-packages`` directory (and any other ``sys.path`` directories) FIRST.

If you are upgrading a previous version of setuptools that was installed using an ``.exe`` installer, please be sure to also **uninstall that older version** via your system's "Add/Remove Programs" feature, BEFORE installing the newer version.

Once installation is complete, you will find an ``easy\_install.exe`` program in your Python ``Scripts`` subdirectory. Be sure to add this directory to your ``PATH`` environment variable, if you haven't already done so.

RPM-Based Systems
=====

Install setuptools using the provided source RPM. The included ``.spec`` file assumes you are installing using the default ``python`` executable, and is not

specific to a particular Python version. The ``easy\_install`` executable will be installed to a system ``bin`` directory such as ``/usr/bin``.

If you wish to install to a location other than the default Python installation's default ``site-packages`` directory (and ``\$prefix/bin`` for scripts), please use the ``.egg``-based installation approach described in the following section.

Cygwin, Mac OS X, Linux, Other

=====

1. Download the appropriate egg for your version of Python (e.g. ``setuptools-0.6c9-py2.4.egg``). Do NOT rename it.
2. Run it as if it were a shell script, e.g. ``sh setuptools-0.6c9-py2.4.egg``. Setuptools will install itself using the matching version of Python (e.g. ``python2.4``), and will place the ``easy\_install`` executable in the default location for installing Python scripts (as determined by the standard distutils configuration files, or by the Python installation).

If you want to install setuptools to somewhere other than ``site-packages`` or your default distutils installation locations for libraries and scripts, you may include EasyInstall command-line options such as ``--prefix``, ``--install-dir``, and so on, following the ``.egg`` filename on the same command line. For example::

```
sh setuptools-0.6c9-py2.4.egg --prefix=~
```

You can use ``--help`` to get a full options list, but we recommend consulting the `EasyInstall manual`_ for detailed instructions, especially the section on custom installation locations`_.

.. _EasyInstall manual: <http://peak.telecommunity.com/DevCenter/EasyInstall>
.. _the section on custom installation locations: <http://peak.telecommunity.com/DevCenter/EasyInstall#custom-installation-locations>

Cygwin Note

If you are trying to install setuptools for the **Windows** version of Python (as opposed to the Cygwin version that lives in ``/usr/bin``), you must make sure that an appropriate executable (``python2.3``, ``python2.4``, or ``python2.5``) is on your **Cygwin** ``PATH`` when invoking the egg. For example, doing the following at a Cygwin bash prompt will install setuptools

for the ****Windows**** Python found at ``C:\Python24``:

```
ln -s /cygdrive/c/Python24/python.exe python2.4
PATH=.:$PATH sh setuptools-0.6c9-py2.4.egg
rm python2.4
```

Downloads

=====

All setuptools downloads can be found at the project's home page in the Python Package Index`. Scroll to the very bottom of the page to find the links.

.. the project's home page in the Python Package Index: <http://pypi.org/pypi/setuptools>

In addition to the PyPI downloads, the development version of ``setuptools`` is available from the `Python SVN sandbox`, and in-development versions of the `0.6 branch` are available as well.

.. 0.6 branch: <http://svn.python.org/projects/sandbox/branches/setuptools-0.6/#egg=setuptools-dev06>

.. Python SVN sandbox: <http://svn.python.org/projects/sandbox/trunk/setuptools/#egg=setuptools-dev>

Using

Setuptools and EasyInstall

Here are some of the available manuals, tutorials, and other resources for learning about Setuptools, Python Eggs, and EasyInstall:

- * `The EasyInstall user's guide and reference manual`_
- * `The setuptools Developer's Guide`_
- * `The pkg\_resources API reference`_
- * `Package Compatibility Notes`_ (user-maintained)
- * `The Internal Structure of Python Eggs`_

Questions, comments, and bug reports should be directed to the `distutils-sig mailing list`. If you have written (or know of) any tutorials, documentation, plug-ins, or other resources for setuptools users, please let us know about them there, so this reference list can be updated. If you have working, *tested* patches to correct problems or add features, you may submit them to the `setuptools bug tracker`_.

.. setuptools bug tracker: <http://bugs.python.org/setuptools/>

.. Package Compatibility Notes: <http://peak.telecommunity.com/DevCenter/PackageNotes>

.. The Internal Structure of Python Eggs: <http://peak.telecommunity.com/DevCenter/EggFormats>

.. The setuptools Developer's Guide: <http://peak.telecommunity.com/DevCenter/setuptools>

.. _The pkg_resources API reference: <http://peak.telecommunity.com/DevCenter/PkgResources>
.. _The EasyInstall user's guide and reference manual: <http://peak.telecommunity.com/DevCenter/EasyInstall>
.. _distutils-sig mailing list: <http://mail.python.org/pipermail/distutils-sig/>

Credits

* The original design for the ``egg`` format and the ``pkg\_resources`` API was co-created by Phillip Eby and Bob Ippolito. Bob also implemented the first version of ``pkg\_resources``, and supplied the OS X operating system version compatibility algorithm.

* Ian Bicking implemented many early "creature comfort" features of easy_install, including support for downloading via Sourceforge and Subversion repositories. Ian's comments on the Web-SIG about WSGI application deployment also inspired the concept of "entry points" in eggs, and he has given talks at PyCon and elsewhere to inform and educate the community about eggs and setuptools.

* Jim Fulton contributed time and effort to build automated tests of various aspects of ``easy\_install``, and supplied the doctests for the command-line ``.exe`` wrappers on Windows.

* Phillip J. Eby is the principal author and maintainer of setuptools, and first proposed the idea of an importable binary distribution format for Python application plug-ins.

* Significant parts of the implementation of setuptools were funded by the Open Source Applications Foundation, to provide a plug-in infrastructure for the Chandler PIM application.

In addition, many OSAF staffers (such as Mike "Code Bear" Taylor) contributed their time and stress as guinea pigs for the use of eggs and setuptools, even before eggs were "cool". (Thanks, guys!)

Keywords: CPAN PyPI distutils eggs package management

Platform: UNKNOWN

Classifier: Development Status :: 3 - Alpha

Classifier: Intended Audience :: Developers

Classifier: License :: OSI Approved :: Python Software Foundation License

Classifier: License :: OSI Approved :: Zope Public License

Classifier: Operating System :: OS Independent

Classifier: Programming Language :: Python

Classifier: Topic :: Software Development :: Libraries :: Python Modules

Classifier: Topic :: System :: Archiving :: Packaging

Classifier: Topic :: System :: Systems Administration

Classifier: Topic :: Utilities

Copyright (C) 2013 by Test User.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Vinay Sajip not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

VINAY SAJIP DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL VINAY SAJIP BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

BSD-style without advertising clause

UNRECOGNIZED LICENSE; MD5 sum: e2ceb64456e606a09824fbf3dd9a158e

UNRECOGNIZED LICENSE; MD5 sum: 39cf6a4b85f545a48318dca445f2d0e6

BSD-licensed.

UNRECOGNIZED LICENSE; MD5 sum: dc4075889a5c2ece34c05148a2401013

UNRECOGNIZED LICENSE; MD5 sum: ee664a1e97eefe77d34ef50e74b7326d

1.2 ipset 7.24-r0

1.2.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below,

refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in

whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but

does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for

making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6.

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot

distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of

this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands

`show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the

library. If this is what you want to do, use the GNU Library General Public License instead of this License.

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who
decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library,
whether gratis
or for a fee, you must give the recipients all the rights that we gave
you. You must make sure that they, too, receive or can get the source
code. If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them

with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes

a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public

License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or

derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a

work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
- b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the

Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally

distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying

the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus

excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF

SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

1.3 bridgeutils 1.7.1-r2

1.3.1 Available under license :

```
/*
 * Copyright (C) 2000 Lennert Buytenhek
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU General Public License as
 * published by the Free Software Foundation; either version 2 of the
 * License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful, but
 * WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
 * General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
 */
```

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for

this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in

the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's

source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program,

and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include

anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution

of the Program by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under
any particular circumstance, the balance of the section is intended to
apply and the section as a whole is intended to apply in other
circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system, which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through
any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies a version number
of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

10. If you wish to incorporate parts of the Program into other free

programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.

EXCEPT WHEN

OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN

IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify

it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License,
or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show
c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.

1.4 typeguard 4.3.0

1.4.1 Available under license :

This is the MIT license: <http://www.opensource.org/licenses/mit-license.php>

Copyright (c) Alex Grnholm

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.5 util-linux 2.41-r9

1.5.1 Available under license :

```
NR START  END SECTORS SIZE NAME UUID
1  32 7679   7648 3.7M   8f8378c0-01
2 7680 16383   8704 4.3M   8f8378c0-02
5 7936 12799   4864 2.4M
6 12544 16127   3584 1.8M
# Automatically generated by apkbuild-cpan, template 4
# Contributor: Valery Kartel <valery.kartel@gmail.com>
# Maintainer: Celeste <cielesti@protonmail.com>
maintainer="Celeste <cielesti@protonmail.com>"
pkgname=perl-bsd-resource
pkgver=1.2911
pkgrel=10
#_pkgreal is used by apkbuild-cpan to find modules at MetaCpan
_pkgreal=BSD-Resource
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="https://metacpan.org/release/BSD-Resource/"
```

```

arch="all"
license="Artistic-2.0 OR LGPL-2.0"
depends="perl"
makedepends="perl-dev"
subpackages="$pkgname-doc"
source="https://cpan.metacpan.org/authors/id/J/JH/JHI/BSD-Resource-$pkgver.tar.gz"
builddir="$srcdir/$_pkgreal-$pkgver"

build() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    PERL_MM_USE_DEFAULT=1 perl -I. Makefile.PL \
        INSTALLDIRS=vendor \
        NO_PACKLIST=1 \
        NO_PERLLOCAL=1
    make
}

check() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    make test
}

package() {
    make DESTDIR="$pkgdir" install
}

sha512sums="
d0032d41c7c0468ed1c6d8f57b885f6cb97a5039d754c8cb60b2067daedaf53bd15fb6561a3d0f828df16dfa5417f663b
8065ba65f4fea16dc9262728b3b6b85
    BSD-Resource-1.2911.tar.gz
"

```

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<https://fsf.org/>>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
 software and other kinds of works.

The licenses for most software and other practical works are designed
 to take away your freedom to share and change the works. By contrast,
 the GNU General Public License is intended to guarantee your freedom to
 share and change all versions of a program--to make sure it remains free
 software for all its users. We, the Free Software Foundation, use the

GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1)

assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand

ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of

software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If

the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The

"System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is

covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed;

section 10

makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately

publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey,
and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to
produce it from the Program, in the form of source code under the
terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified
it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is
released under this License and any conditions added under section
7. This requirement modifies the requirement in section
4 to
"keep intact all notices".
- c) You must license the entire work, as a whole, under this
License to anyone who comes into possession of a copy. This
License will therefore apply, along with any applicable section 7
additional terms, to the whole of the work, and all its parts,
regardless of how they are packaged. This License gives no
permission to license the work in any other way, but it does not
invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display
Appropriate Legal Notices; however, if the Program has interactive
interfaces that do not display Appropriate Legal Notices, your
work need not make them do so.

A compilation of a covered work with other separate and independent
works, which are not by their nature extensions of the covered work,
and which are not combined with it such as to form a larger program,
in or on a volume of a storage or distribution
medium, is called an
"aggregate" if the compilation and its resulting copyright are not
used to limit the access or legal rights of the compilation's users
beyond what the individual works permit. Inclusion of a covered work
in an aggregate does not cause this License to apply to the other
parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no

charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright

holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered

work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),

EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17.

Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>

This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary.

For more information on this, and how to apply and follow the GNU GPL, see
<<https://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want

to do, use the GNU Lesser General
Public License instead of this License. But first, please read
<<https://www.gnu.org/licenses/why-not-lgpl.html>>.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next paragraph) shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE
OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, and the entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the

documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ALL OF WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF NOT ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Contributor: Maxim Karasev <begs@disroot.org>

Maintainer: mio <miyopan@e.email>

pkgname=bsd-games

pkgver=3.3

pkgrel=2

pkgdesc="traditional text mode games from BSD"

url="https://bsd-games.sourceforge.io/"

arch="all"

license="BSD-3-Clause"

install="\$pkgname.post-install"

makedepends="ncurses-dev coreutils" # configure causes busybox expr error

subpackages="\$pkgname-doc"

source="https://sourceforge.net/projects/bsd-games/files/bsd-games-\$pkgver.tar.gz
0001-bsd-games-scores-dir.patch
"

options="!check" # no tests

build() {

some GNU autoconf options are ignored, but it works

./configure \

--build=\$CBUILD \

--host=\$CHOST \

--prefix=/usr \

--sysconfdir=/etc \

--mandir=/usr/share/man \

--sourcedir=/usr/share/\$pkgname

make

}

package() {

```

make DESTDIR="$pkgdir" install
install -Dm644 LICENSE "$pkgdir"/usr/share/licenses/bsd-games/LICENSE
}

sha512sums="
aaf36d09d4fe68514a5c279063d7e77a9a9a84c447037cba974b29faa07eb80c2aab2379d0699c196b5c27ffe1b2bb2c6
8d3390143e0e62d5e3d210ccfb61294
  bsd-games-3.3.tar.gz
e1def0d4835a877242c4173fae862e278a7858738e0bacacbb045a42bcabec2aa7c72d000bbd2421ce35fe66cdc1f52c0
577b7e278029ca5ed50f7aab22319f6 0001-bsd-games-scores-dir.patch
"

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
# Maintainer: Natanael Copa <ncopa@alpinelinux.org>
pkgname=libart-lgpl
pkgver=2.3.21
pkgrel=8
pkgdesc="A library for high-performance 2D graphics"
url="https://www.levien.com/libart/"
arch="all"
options="!check" # No test suite.
license="LGPL-2.0-or-later"
subpackages="$pkgname-dev"
source="https://download.gnome.org/sources/libart-lgpl/2.3/libart-lgpl-$pkgver.tar.bz2"
"

builddir="$srcdir"/libart-lgpl-$pkgver

prepare() {
  update_config_sub
  default_prepare
}

build() {
  ./configure \
  --build=$CBUILD \
  --host=$CHOST \
  --prefix=/usr
  make

```

```
}
```

```
package() {  
  make DESTDIR="$pkgdir" install  
}
```

```
sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7  
6654309e001aee9bace727b5b4c8589405256a3c020 libart_lgpl-2.3.21.tar.bz2"
```

```
mini_sendmail - accept email on behalf of real sendmail
```

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

The complete text of the license is available in the
../Documentation/licenses/COPYING.LGPL-2.1-or-later file.

```
/*
```

```
* Copyright (c) 1989 The Regents of the University of California.
```

```
* All rights reserved.
```

```
*
```

```
* Redistribution and use in source and binary forms, with or without
```

```
* modification, are permitted provided that the following conditions
```

* are met:

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. All advertising materials mentioning features or use of this software

* must display the following acknowledgement:

* This product includes software developed by the University of

* California, Berkeley and its contributors.

* 4. Neither the name of the University nor the names of its contributors

* may be used to endorse or promote products derived from this software

*

without specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND

* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE

* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS

* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT

* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY

* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF

* SUCH DAMAGE.

*/

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

HOWEVER

CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```

# Contributor: Carlo Landmeter <clandmeter@alpinelinux.org>
# Maintainer: Kevin Daudt <kdaudt@alpinelinux.org>
pkgname=spdx-licenses
pkgver=3.26.0
pkgrel=0
pkgdesc="Various data formats for the SPDX License List"
url="https://spdx.org/"
arch="noarch"
options="!check" # no test suite
license="CC-BY-3.0"
source="license-list-data-$pkgver.tar.gz::https://github.com/spdx/license-list-data/archive/v$pkgver.tar.gz"
builddir="$srcdir/license-list-data-$pkgver"
subpackages="$pkgname-list"

_types="html json rdfa rdft rdfturtle rdfxml template text"

for type in $_types; do
    subpackages="$subpackages $pkgname-$type:_subpkg"
done

package() {
    mkdir -p "$pkgdir"
}

_subpkg() {
    local type=${subpkgname/$pkgname-/}
    pkgdesc="$pkgdesc ($type)"
    install_if="$pkgname=$pkgver-r$pkgrel"
    mkdir -p "$subpkgdir"/usr/share/spdx
    cp -r "$builddir"/$type "$subpkgdir"/usr/share/spdx/
}

list() {
    pkgdesc="$pkgdesc (licence list)"
    mkdir -p "$subpkgdir"/usr/share/spdx
    local i; for i in "$builddir"/text/*.txt;
    do
        local license=${i##*/}
        echo ${license%.*} >> "$subpkgdir"/usr/share/spdx/license.lst
    done
}

sha512sums="
67e618d4642dfe4f366935d9514d6f7941d3711e77c49bc1b340ac572d36015981b8f10c72a9f80f1d83e132ae7e5093c
0155e0c0e786e4777ee16d1cb7b885a license-list-data-3.26.0.tar.gz
"

Original Copyright (c) 1985 by Supoj Sutanthavibul
Parts Copyright (c) 1989-2012 by Brian V. Smith

```


Parts Copyright (c) 1991 by Paul King
Other Copyrights may be found in various files

Main Xfig copyright notice:

Any party obtaining a copy of these files is granted, free of charge, a full and unrestricted irrevocable, world-wide, paid up, royalty-free, nonexclusive right and license to deal in this software and documentation files (the "Software"), including without limitation the rights to use, copy, modify, merge, publish and/or distribute copies of the Software, and to permit persons who receive copies from any such party to do so, with the only requirement being that this copyright notice remain intact.

No representations are made about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Parts Copyright (C) 1993 by Alan Richardson (mppa3@uk.ac.sussex.syms)
The text rotation code in w_rottext.c was written by Alan Richardson.
The above copyright notice holds for this work as well.

Parts Copyright (c) 1994 by Anthony Dekker
The Kohonen neural network code for color optimization was written by Anthony Dekker.

Parts Copyright (c) 1995 by C. Blanc and C. Schlick
The X-Spline code was written Carole Blanc (blanc@labri.u-bordeaux.fr) and Christophe Schlick (schlick@labri.u-bordeaux.fr) starting from an initial implementation done by C. Feuille, S. Grosbois, L. Maziere and L. Minihot as a student practice (Universite Bordeaux, France). For additional information about X-splines, see:
"X-Splines: A Spline Model Designed for the End User" by C. Blanc and C. Schlick, Proceedings of SIGGRAPH'95
<http://dept-info.labri.u-bordeaux.fr/~schlick/DOC/sig1.html>
Contributor: Dhruvin Gandhi <contact@dhruvin>
Maintainer: Patrycja Rosa <alpine@ptrcnull.me>
pkgname=py3-license-expression
_pkgname=license_expression
pkgver=30.4.1
pkgrel=0
pkgdesc="Library to parse, compare, simplify and normalize license expressions"
url="https://github.com/nexB/license-expression"
arch="noarch"
license="Apache-2.0"
depends="python3 py3-boolean.py"
makedepends="py3-setuptools py3-setuptools_scm py3-gpep517 py3-wheel"
checkdepends="py3-pytest py3-pytest-xdist"
subpackages="\$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/l/_pkgname/_pkgname-\$pkgver.tar.gz"
builddir="\$srcdir/\$pkgname-\$pkgver"

```
build() {  
    gpep517 build-wheel \  
        --wheel-dir .dist \  
        --output-fd 3 3>&1 >&2  
}
```

```

}

check() {
python3 -m venv --clear --without-pip --system-site-packages .testenv
.testenv/bin/python3 -m installer .dist/*.whl
.testenv/bin/python3 -m pytest
}

package() {
python3 -m installer -d "$pkgdir" \
.dists/*.whl
}

sha512sums="
114c630f098939757e49187e35b4e54883266a2ee9ca283724992420ef4c860f9606a8342da0ab74d93627c10805f312
fda3f699cbc7c4fbd47a71e220440fb0
license_expression-30.4.1.tar.gz
"

```

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program).

Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the

entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as

distribution of the source
code, even though third parties are not
compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing
or modifying
the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For
example, if a patent
license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE,
THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR
THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU
General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute
it
under certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be

reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those

sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public

License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you

distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or

otherwise) that contradict the conditions of this License, they do not excuse

you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system

which is

implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by

the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
Contributor: Fabian Affolter <fabian@affolter-engineering.ch>
Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>
pkgname=py3-flake8-copyright
_pkgname=flake8-copyright
pkgver=0.2.4
pkgrel=3
pkgdesc="Extension for flake8 which checks for copyrights"
options="!check" # No testsuite
url="https://github.com/savoirfairelinux/flake8-copyright"
arch="noarch"
license="MIT"

```

depends="py3-flake8 py3-setuptools"
makedepends="py3-gpep517 py3-wheel"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/${_pkgname:0:1}/${_pkgname}/${_pkgname}-${pkgver}.tar.gz"
builddir="$srcdir"/$_pkgname-$pkgver

replaces="py-flake8-copyright"          # Backwards compatibility
provides="py-flake8-copyright=$pkgver-r$pkgrel" # Backwards compatibility

build() {
    gpep517 build-wheel \
        --wheel-dir .dist \
        --output-fd 3 3>&1 >&2
}

package() {
    gpep517 install-wheel --destdir "$pkgdir" \
        .dist/*.whl
}

sha512sums="
a6ab47e1bb715618f075c51f398e18180404871b3b0faf1c2d30701d5203db6bc23771bf22ffb5bafcd8ee856b9b64237
b316bd503ae7dceed6ca284ccd5a74d
    flake8-copyright-0.2.4.tar.gz
"

/* $NetBSD: tree.h,v 1.20 2013/09/14 13:20:45 joerg Exp $ */
/* $OpenBSD: tree.h,v 1.13 2011/07/09 00:19:45 pirofti Exp $ */
/* Modified by Void Linux. */
/*
* Copyright 2002 Niels Provos <provos@citi.umich.edu>
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE
* AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

```

```

* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```

```

#ifndef _SYS_TREE_H_
#define _SYS_TREE_H_

```

```

#ifdef __GNUC__
#define __GNUC_PREREQ__(x, y) \
    ((__GNUC__ == (x) && __GNUC_MINOR__ >= (y)) || \
     (__GNUC__ > (x)))
#else
#define __GNUC_PREREQ__(x, y) 0
#endif

```

```

#if __GNUC_PREREQ__(2, 7) || defined(__lint__)
#define _sys_tree_h_unused __attribute__((__unused__))
#else
#define _sys_tree_h_unused /* delete */
#endif

```

```

/*

```

```

* This file defines data structures for different types of trees:
* splay trees and red-black trees.
*
* A splay tree is a self-organizing
* data structure. Every operation
* on the tree causes a splay to happen. The splay moves the requested
* node to the root of the tree and partly rebalances it.
*
* This has the benefit that request locality causes faster lookups as
* the requested nodes move to the top of the tree. On the other hand,
* every lookup causes memory writes.
*
* The Balance Theorem bounds the total access time for m operations
* and n inserts on an initially empty tree as  $O((m + n) \lg n)$ . The
* amortized cost for a sequence of m accesses to a splay tree is  $O(\lg n)$ ;
*
* A red-black tree is a binary search tree with the node color as an
* extra attribute. It fulfills a set of conditions:
* - every search path from the root to a leaf consists of the
*   same number of black nodes,
* - each red node (except for the root) has a black parent,
* - each leaf node is black.
*
* Every operation on a red-black tree is bounded as  $O(\lg n)$ .

```

* The maximum height of a red-black tree is $2\lg(n+1)$.

*/

```
#define SPLAY_HEAD(name, type)  \
struct name {                  \
    struct type *sph_root; /* root of the tree */ \
}
```

```
#define SPLAY_INITIALIZER(root)  \
{ NULL }
```

```
#define SPLAY_INIT(root) do { \
    (root)->sph_root = NULL; \
} while (/*CONSTCOND*/ 0)
```

```
#define SPLAY_ENTRY(type)  \
struct {                   \
    struct type *spe_left; /* left element */ \
    struct type *spe_right; /* right element */ \
}
```

```
#define SPLAY_LEFT(elm, field) (elm)->field.spe_left
#define SPLAY_RIGHT(elm, field) (elm)->field.spe_right
#define SPLAY_ROOT(head) (head)->sph_root
#define SPLAY_EMPTY(head) (SPLAY_ROOT(head) == NULL)
```

```
/* SPLAY_ROTATE_{LEFT,RIGHT} expect that tmp hold SPLAY_{RIGHT,LEFT} */
#define SPLAY_ROTATE_RIGHT(head, tmp, field) do { \
    SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(tmp, field); \
    SPLAY_RIGHT(tmp, field) = (head)->sph_root; \
    (head)->sph_root = tmp; \
} while (/*CONSTCOND*/ 0)
```

```
#define SPLAY_ROTATE_LEFT(head, tmp, field) do { \
    SPLAY_RIGHT((head)->sph_root, \
    field) = SPLAY_LEFT(tmp, field); \
    SPLAY_LEFT(tmp, field) = (head)->sph_root; \
    (head)->sph_root = tmp; \
} while (/*CONSTCOND*/ 0)
```

```
#define SPLAY_LINKLEFT(head, tmp, field) do { \
    SPLAY_LEFT(tmp, field) = (head)->sph_root; \
    tmp = (head)->sph_root; \
    (head)->sph_root = SPLAY_LEFT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)
```

```
#define SPLAY_LINKRIGHT(head, tmp, field) do { \
    SPLAY_RIGHT(tmp, field) = (head)->sph_root; \
}
```

```

tmp = (head)->sph_root;    \
(head)->sph_root = SPLAY_RIGHT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ASSEMBLE(head, node, left, right, field) do { \
    SPLAY_RIGHT(left, field) = SPLAY_LEFT((head)->sph_root, field); \
    SPLAY_LEFT(right, field) = SPLAY_RIGHT((head)->sph_root, field); \
    SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(node, field); \
    SPLAY_RIGHT((head)->sph_root, field) = SPLAY_LEFT(node, field); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */

#define
SPLAY_PROTOTYPE(name, type, field, cmp) \
void name##_SPLAY(struct name *, struct type *); \
void name##_SPLAY_MINMAX(struct name *, int); \
struct type *name##_SPLAY_INSERT(struct name *, struct type *); \
struct type *name##_SPLAY_REMOVE(struct name *, struct type *); \
\
/* Finds the node with the same key as elm */ \
static __inline struct type * \
name##_SPLAY_FIND(struct name *head, struct type *elm) \
{ \
    if (SPLAY_EMPTY(head)) \
        return(NULL); \
    name##_SPLAY(head, elm); \
    if ((cmp)(elm, (head)->sph_root) == 0) \
        return (head->sph_root); \
    return (NULL); \
} \
\
static __inline _sys_tree_h_unused struct type * \
name##_SPLAY_NEXT(struct name *head, struct type *elm) \
{ \
    name##_SPLAY(head, elm); \
    if (SPLAY_RIGHT(elm, field) != NULL) { \
        elm = SPLAY_RIGHT(elm, field); \
        while (SPLAY_LEFT(elm, field) != NULL) { \
            elm = SPLAY_LEFT(elm, field); \
        } \
    } \
    else \
        elm = NULL; \
    return (elm); \
} \
\
static _sys_tree_h_unused __inline struct type * \

```



```

name##_SPLAY_MIN_MAX(struct name *head, int val) \
{ \
    name##_SPLAY_MINMAX(head, val); \
    return (SPLAY_ROOT(head)); \
}

/* Main splay operation.
 * Moves node close to the key of elm to top
 */
#define SPLAY_GENERATE(name, type, field, cmp) \
struct type * \
name##_SPLAY_INSERT(struct name *head, struct type *elm) \
{ \
    if (SPLAY_EMPTY(head)) { \
        SPLAY_LEFT(elm, field) = SPLAY_RIGHT(elm, field) = NULL; \
    } else { \
        int __comp; \
        name##_SPLAY(head, elm); \
        __comp = (cmp)(elm, (head)->sph_root); \
        if (__comp < 0) { \
            SPLAY_LEFT(elm, field) = SPLAY_LEFT((head)->sph_root, field); \
            SPLAY_RIGHT(elm, field) = (head)->sph_root; \
            SPLAY_LEFT((head)->sph_root, field) = NULL; \
        } else if (__comp > 0) { \
            SPLAY_RIGHT(elm, \
field) = SPLAY_RIGHT((head)->sph_root, field); \
            SPLAY_LEFT(elm, field) = (head)->sph_root; \
            SPLAY_RIGHT((head)->sph_root, field) = NULL; \
        } else \
            return ((head)->sph_root); \
    } \
    (head)->sph_root = (elm); \
    return (NULL); \
} \
\
struct type * \
name##_SPLAY_REMOVE(struct name *head, struct type *elm) \
{ \
    struct type *__tmp; \
    if (SPLAY_EMPTY(head)) \
        return (NULL); \
    name##_SPLAY(head, elm); \
    if ((cmp)(elm, (head)->sph_root) == 0) { \
        if (SPLAY_LEFT((head)->sph_root, field) == NULL) { \
            (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field); \
        } else { \
            __tmp = SPLAY_RIGHT((head)->sph_root, field); \
            (head)->sph_root = SPLAY_LEFT((head)->sph_root, field); \

```

```

    name##_SPLAY(head, elm); \
    SPLAY_RIGHT((head)->sph_root, field) = __tmp; \
} \
return (elm); \
} \
return (NULL); \
} \
\
void \
name##_SPLAY(struct
name *head, struct type *elm) \
{ \
    struct type __node, *__left, *__right, *__tmp; \
    int __comp; \
\
    SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
    __left = __right = &__node; \
\
    while ((__comp = (cmp)(elm, (head)->sph_root)) != 0) { \
        if (__comp < 0) { \
            __tmp = SPLAY_LEFT((head)->sph_root, field); \
            if (__tmp == NULL) \
                break; \
            if ((cmp)(elm, __tmp) < 0){ \
                SPLAY_ROTATE_RIGHT(head, __tmp, field); \
                if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
                    break; \
            } \
            SPLAY_LINKLEFT(head, __right, field); \
        } else if (__comp > 0) { \
            __tmp = SPLAY_RIGHT((head)->sph_root, field); \
            if (__tmp == NULL) \
                break; \
            if ((cmp)(elm, __tmp) > 0){ \
                SPLAY_ROTATE_LEFT(head, __tmp, field); \
                if (SPLAY_RIGHT((head)->sph_root, field) == NULL)\
                    break; \
            } \
            SPLAY_LINKRIGHT(head, __left, field); \
        } \
    } \
    SPLAY_ASSEMBLE(head,
    &__node, __left, __right, field); \
} \
\
/* Splay with either the minimum or the maximum element \
* Used to find minimum or maximum element in tree. \
*/ \

```

```

void name##_SPLAY_MINMAX(struct name *head, int __comp) \
{ \
    struct type __node, *__left, *__right, *__tmp; \
\
    SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
    __left = __right = &__node; \
\
    while (1) { \
        if (__comp < 0) { \
            __tmp = SPLAY_LEFT((head)->sph_root, field);\
            if (__tmp == NULL) \
                break; \
            if (__comp < 0){ \
                SPLAY_ROTATE_RIGHT(head, __tmp, field);\
                if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
                    break; \
            } \
            SPLAY_LINKLEFT(head, __right, field); \
        } else if (__comp > 0) { \
            __tmp = SPLAY_RIGHT((head)->sph_root, field);\
            if (__tmp == NULL) \
                break; \
            if (__comp > 0) { \
                SPLAY_ROTATE_LEFT(head, __tmp, field);\
                if (SPLAY_RIGHT((head)->sph_root, \
field) == NULL)\
                    break; \
            } \
            SPLAY_LINKRIGHT(head, __left, field); \
        } \
    } \
    SPLAY_ASSEMBLE(head, &__node, __left, __right, field); \
}

```

```

#define SPLAY_NEGINF -1

```

```

#define SPLAY_INF 1

```

```

#define SPLAY_INSERT(name, x, y) name##_SPLAY_INSERT(x, y)

```

```

#define SPLAY_REMOVE(name, x, y) name##_SPLAY_REMOVE(x, y)

```

```

#define SPLAY_FIND(name, x, y) name##_SPLAY_FIND(x, y)

```

```

#define SPLAY_NEXT(name, x, y) name##_SPLAY_NEXT(x, y)

```

```

#define SPLAY_MIN(name, x) (SPLAY_EMPTY(x) ? NULL \

```

```

    : name##_SPLAY_MIN_MAX(x, SPLAY_NEGINF))

```

```

#define SPLAY_MAX(name, x) (SPLAY_EMPTY(x) ? NULL \

```

```

    : name##_SPLAY_MIN_MAX(x, SPLAY_INF))

```

```

#define SPLAY_FOREACH(x, name, head) \

```

```

    for ((x) = SPLAY_MIN(name, head); \

```

```

(x) != NULL;    \
(x) = SPLAY_NEXT(name, head, x))

/* Macros that define a red-black tree */
#define RB_HEAD(name, type)    \
struct name {                  \
    struct type *rbh_root; /* root of the tree */ \
}

#define RB_INITIALIZER(root)    \
{                                \
    NULL }

#define RB_INIT(root) do {      \
    (root)->rbh_root = NULL;    \
} while (/*CONSTCOND*/ 0)

#define RB_BLACK 0
#define RB_RED 1
#define RB_ENTRY(type)        \
struct {                       \
    struct type *rbe_left; /* left element */ \
    struct type *rbe_right; /* right element */ \
    struct type *rbe_parent; /* parent element */ \
    int rbe_color; /* node color */ \
}

#define RB_LEFT(elm, field) (elm)->field.rbe_left
#define RB_RIGHT(elm, field) (elm)->field.rbe_right
#define RB_PARENT(elm, field) (elm)->field.rbe_parent
#define RB_COLOR(elm, field) (elm)->field.rbe_color
#define RB_ROOT(head) (head)->rbh_root
#define RB_EMPTY(head) (RB_ROOT(head) == NULL)

#define RB_SET(elm, parent, field) do { \
    RB_PARENT(elm, field) = parent; \
    RB_LEFT(elm, field) = RB_RIGHT(elm, field) = NULL; \
    RB_COLOR(elm, field) = RB_RED; \
} while (/*CONSTCOND*/ 0)

#define RB_SET_BLACKRED(black, red, field) do { \
    RB_COLOR(black, field) = RB_BLACK; \
    RB_COLOR(red, field) \
    = RB_RED; \
} while (/*CONSTCOND*/ 0)

#ifndef RB_AUGMENT
#define RB_AUGMENT(x) do {} while (/*CONSTCOND*/ 0)

```

```

#endif

#define RB_ROTATE_LEFT(head, elm, tmp, field) do { \
    (tmp) = RB_RIGHT(elm, field); \
    if ((RB_RIGHT(elm, field) = RB_LEFT(tmp, field)) != NULL) { \
        RB_PARENT(RB_LEFT(tmp, field), field) = (elm); \
    } \
    RB_AUGMENT(elm); \
    if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
        if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
            RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
        else \
            RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
    } else \
        (head)->rbh_root = (tmp); \
    RB_LEFT(tmp, field) = (elm); \
    RB_PARENT(elm, field) = (tmp); \
    RB_AUGMENT(tmp); \
    if ((RB_PARENT(tmp, field))) \
        RB_AUGMENT(RB_PARENT(tmp, field)); \
} while (/*CONSTCOND*/ 0)

#define RB_ROTATE_RIGHT(head, elm, tmp, field) do { \
    (tmp) = RB_LEFT(elm, field); \
    if ((RB_LEFT(elm, field) = RB_RIGHT(tmp, field)) \
        != NULL) { \
        RB_PARENT(RB_RIGHT(tmp, field), field) = (elm); \
    } \
    RB_AUGMENT(elm); \
    if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
        if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
            RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
        else \
            RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
    } else \
        (head)->rbh_root = (tmp); \
    RB_RIGHT(tmp, field) = (elm); \
    RB_PARENT(elm, field) = (tmp); \
    RB_AUGMENT(tmp); \
    if ((RB_PARENT(tmp, field))) \
        RB_AUGMENT(RB_PARENT(tmp, field)); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */
#define RB_PROTOTYPE(name, type, field, cmp) \
    RB_PROTOTYPE_INTERNAL(name, type, field, cmp,)
#define RB_PROTOTYPE_STATIC(name, type, field, cmp) \
    RB_PROTOTYPE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)

```

```

#define RB_PROTOTYPE_INTERNAL(name, type, field, cmp, attr) \
attr void name##_RB_INSERT_COLOR(struct name *, struct \
type *); \
attr void name##_RB_REMOVE_COLOR(struct name *, struct type *, struct type *);\
attr struct type *name##_RB_REMOVE(struct name *, struct type *); \
attr struct type *name##_RB_INSERT(struct name *, struct type *); \
attr struct type *name##_RB_FIND(struct name *, struct type *); \
attr struct type *name##_RB_NFIND(struct name *, struct type *); \
attr struct type *name##_RB_NEXT(struct type *); \
attr struct type *name##_RB_PREV(struct type *); \
attr struct type *name##_RB_MINMAX(struct name *, int); \
\

/* Main rb operation.
* Moves node close to the key of elm to top
*/
#define RB_GENERATE(name, type, field, cmp) \
RB_GENERATE_INTERNAL(name, type, field, cmp,)
#define RB_GENERATE_STATIC(name, type, field, cmp) \
RB_GENERATE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_GENERATE_INTERNAL(name, type, field, cmp, attr) \
attr void \
name##_RB_INSERT_COLOR(struct name *head, struct type *elm) \
{ \
struct type \
*parent, *gparent, *tmp; \
while ((parent = RB_PARENT(elm, field)) != NULL && \
RB_COLOR(parent, field) == RB_RED) { \
gparent = RB_PARENT(parent, field); \
if (parent == RB_LEFT(gparent, field)) { \
tmp = RB_RIGHT(gparent, field); \
if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
RB_COLOR(tmp, field) = RB_BLACK; \
RB_SET_BLACKRED(parent, gparent, field);\
elm = gparent; \
continue; \
} \
if (RB_RIGHT(parent, field) == elm) { \
RB_ROTATE_LEFT(head, parent, tmp, field);\
tmp = parent; \
parent = elm; \
elm = tmp; \
} \
RB_SET_BLACKRED(parent, gparent, field); \
RB_ROTATE_RIGHT(head, gparent, tmp, field); \
} else { \
tmp = RB_LEFT(gparent, field); \
if (tmp && RB_COLOR(tmp, field) == RB_RED) { \

```

```

    RB_COLOR(tmp, field) = RB_BLACK; \
    RB_SET_BLACKRED(parent, gparent, field);\
    elm = gparent; \
    continue; \
} \
if (RB_LEFT(parent, field) == elm)
{ \
    RB_ROTATE_RIGHT(head, parent, tmp, field);\
    tmp = parent; \
    parent = elm; \
    elm = tmp; \
} \
    RB_SET_BLACKRED(parent, gparent, field); \
    RB_ROTATE_LEFT(head, gparent, tmp, field); \
} \
} \
RB_COLOR(head->rbh_root, field) = RB_BLACK; \
} \
\
attr void \
name##_RB_REMOVE_COLOR(struct name *head, struct type *parent, struct type *elm) \
{ \
    struct type *tmp; \
    while ((elm == NULL || RB_COLOR(elm, field) == RB_BLACK) && \
        elm != RB_ROOT(head)) { \
        if (RB_LEFT(parent, field) == elm) { \
            tmp = RB_RIGHT(parent, field); \
            if (RB_COLOR(tmp, field) == RB_RED) { \
                RB_SET_BLACKRED(tmp, parent, field); \
                RB_ROTATE_LEFT(head, parent, tmp, field);\
                tmp = RB_RIGHT(parent, field); \
            } \
            if ((RB_LEFT(tmp, field) == NULL || \
                RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) && \
                (RB_RIGHT(tmp, field) == NULL || \
                RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK)) {\
                RB_COLOR(tmp, field) = RB_RED; \
                elm = parent; \
                parent = RB_PARENT(elm, field); \
            } else { \
                if (RB_RIGHT(tmp, field) == NULL || \
                    RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK) {\
                    struct type *oleft; \
                    if ((oleft = RB_LEFT(tmp, field)) \
                        != NULL) \
                        RB_COLOR(oleft, field) = RB_BLACK;\
                    RB_COLOR(tmp, field) = RB_RED; \

```

```

    RB_ROTATE_RIGHT(head, tmp, oleft, field);\
    tmp = RB_RIGHT(parent, field);\
} \
RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
RB_COLOR(parent, field) = RB_BLACK;\
if (RB_RIGHT(tmp, field)) \
    RB_COLOR(RB_RIGHT(tmp, field), field) = RB_BLACK;\
RB_ROTATE_LEFT(head, parent, tmp, field);\
elm = RB_ROOT(head); \
break; \
} \
} else { \
    tmp = RB_LEFT(parent, field); \
    if (RB_COLOR(tmp, field) == RB_RED) { \
        RB_SET_BLACKRED(tmp, parent, field);\
        RB_ROTATE_RIGHT(head, parent, tmp, field);\
        tmp
= RB_LEFT(parent, field); \
    } \
    if ((RB_LEFT(tmp, field) == NULL || \
        RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) &&\
        (RB_RIGHT(tmp, field) == NULL || \
        RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK)) {\
        RB_COLOR(tmp, field) = RB_RED; \
        elm = parent; \
        parent = RB_PARENT(elm, field); \
    } else { \
        if (RB_LEFT(tmp, field) == NULL || \
            RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) {\
            struct type *oright; \
            if ((oright = RB_RIGHT(tmp, field)) \
                != NULL) \
                RB_COLOR(oright, field) = RB_BLACK;\
            RB_COLOR(tmp, field) = RB_RED; \
            RB_ROTATE_LEFT(head, tmp, oright, field);\
            tmp = RB_LEFT(parent, field); \
        } \
        RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
        RB_COLOR(parent, field) = RB_BLACK; \
        if (RB_LEFT(tmp, field)) \
            RB_COLOR(RB_LEFT(tmp, field), field) = RB_BLACK;\
        RB_ROTATE_RIGHT(head, parent, tmp, field);\
        elm = RB_ROOT(head); \
        break; \
    } \
} \
} \
} \
if

```



```

(elm)    \
    RB_COLOR(elm, field) = RB_BLACK; \
}      \
    \
attr struct type *    \
name##_RB_REMOVE(struct name *head, struct type *elm) \
{      \
    struct type *child, *parent, *old = elm; \
    int color;    \
    if (RB_LEFT(elm, field) == NULL) \
        child = RB_RIGHT(elm, field); \
    else if (RB_RIGHT(elm, field) == NULL) \
        child = RB_LEFT(elm, field); \
    else {      \
        struct type *left; \
        elm = RB_RIGHT(elm, field); \
        while ((left = RB_LEFT(elm, field)) != NULL) \
            elm = left; \
        child = RB_RIGHT(elm, field); \
        parent = RB_PARENT(elm, field); \
        color = RB_COLOR(elm, field); \
        if (child) \
            RB_PARENT(child, field) = parent; \
        if (parent) { \
            if (RB_LEFT(parent, field) == elm) \
                RB_LEFT(parent, field) = child; \
            else \
                RB_RIGHT(parent, field) = child; \
            RB_AUGMENT(parent); \
        } else \
            RB_ROOT(head)
= child; \
    if (RB_PARENT(elm, field) == old) \
        parent = elm; \
    (elm)->field = (old)->field; \
    if (RB_PARENT(old, field)) { \
        if (RB_LEFT(RB_PARENT(old, field), field) == old)\
            RB_LEFT(RB_PARENT(old, field), field) = elm;\
        else \
            RB_RIGHT(RB_PARENT(old, field), field) = elm;\
        RB_AUGMENT(RB_PARENT(old, field)); \
    } else \
        RB_ROOT(head) = elm; \
    RB_PARENT(RB_LEFT(old, field), field) = elm; \
    if (RB_RIGHT(old, field)) \
        RB_PARENT(RB_RIGHT(old, field), field) = elm; \
    if (parent) { \
        left = parent; \

```

```

do {
    RB_AUGMENT(left);
} while ((left = RB_PARENT(left, field)) != NULL);
}
goto color;
}
parent = RB_PARENT(elm, field);
color = RB_COLOR(elm, field);
if (child)
    RB_PARENT(child, field) = parent;
if (parent) {
    if (RB_LEFT(parent, field) == elm)
        RB_LEFT(parent, field) = child;
    else
        RB_RIGHT(parent,
field) = child;
    RB_AUGMENT(parent);
} else
    RB_ROOT(head) = child;
color:
if (color == RB_BLACK)
    name##_RB_REMOVE_COLOR(head, parent, child);
return (old);
}

/* Inserts a node into the RB tree */
attr struct type *
name##_RB_INSERT(struct name *head, struct type *elm)
{
    struct type *tmp;
    struct type *parent = NULL;
    int comp = 0;
    tmp = RB_ROOT(head);
    while (tmp) {
        parent = tmp;
        comp = (cmp)(elm, parent);
        if (comp < 0)
            tmp = RB_LEFT(tmp, field);
        else if (comp > 0)
            tmp = RB_RIGHT(tmp, field);
        else
            return (tmp);
    }
    RB_SET(elm, parent, field);
    if (parent != NULL) {
        if (comp < 0)
            RB_LEFT(parent, field) = elm;
        else

```

```

    RB_RIGHT(parent, field) = elm; \
    RB_AUGMENT(parent); \
} else \
    RB_ROOT(head) = elm; \
name##_RB_INSERT_COLOR(head,
elm); \
return (NULL); \
} \
\
/* Finds the node with the same key as elm */ \
attr struct type * \
name##_RB_FIND(struct name *head, struct type *elm) \
{ \
    struct type *tmp = RB_ROOT(head); \
    int comp; \
    while (tmp) { \
        comp = cmp(elm, tmp); \
        if (comp < 0) \
            tmp = RB_LEFT(tmp, field); \
        else if (comp > 0) \
            tmp = RB_RIGHT(tmp, field); \
        else \
            return (tmp); \
    } \
    return (NULL); \
} \
\
/* Finds the first node greater than or equal to the search key */ \
attr struct type * \
name##_RB_NFIND(struct name *head, struct type *elm) \
{ \
    struct type *tmp = RB_ROOT(head); \
    struct type *res = NULL; \
    int comp; \
    while (tmp) { \
        comp = cmp(elm, tmp); \
        if (comp < 0) { \
            res = tmp; \
            tmp = RB_LEFT(tmp, field); \
        } \
        else if (comp > 0) \
            tmp = \
RB_RIGHT(tmp, field); \
        else \
            return (tmp); \
    } \
    return (res); \
} \

```

```

    \
/* ARGSUSED */    \
attr struct type *    \
name##_RB_NEXT(struct type *elm)    \
{    \
    if (RB_RIGHT(elm, field)) {    \
        elm = RB_RIGHT(elm, field);    \
        while (RB_LEFT(elm, field))    \
            elm = RB_LEFT(elm, field);    \
    } else {    \
        if (RB_PARENT(elm, field) &&    \
            (elm == RB_LEFT(RB_PARENT(elm, field), field)))    \
            elm = RB_PARENT(elm, field);    \
        else {    \
            while (RB_PARENT(elm, field) &&    \
                (elm == RB_RIGHT(RB_PARENT(elm, field), field)))    \
                elm = RB_PARENT(elm, field);    \
        }    \
    }    \
    return (elm);    \
}    \
    \
/* ARGSUSED */    \
attr struct type *    \
name##_RB_PREV(struct type *elm)    \
{    \
    if (RB_LEFT(elm, field)) {    \
        elm = RB_LEFT(elm, field);    \
        while (RB_RIGHT(elm, field))    \
            elm = RB_RIGHT(elm, field);    \
    }    \
    else {    \
        if (RB_PARENT(elm, field) &&    \
            (elm == RB_RIGHT(RB_PARENT(elm, field), field)))    \
            elm = RB_PARENT(elm, field);    \
        else {    \
            while (RB_PARENT(elm, field) &&    \
                (elm == RB_LEFT(RB_PARENT(elm, field), field)))    \
                elm = RB_PARENT(elm, field);    \
            elm = RB_PARENT(elm, field);    \
        }    \
    }    \
    return (elm);    \
}    \
    \
attr struct type *    \
name##_RB_MINMAX(struct name *head, int val)    \

```

```

{
    \
    struct type *tmp = RB_ROOT(head); \
    struct type *parent = NULL; \
    while (tmp) {
        \
        parent = tmp; \
        if (val < 0) \
            tmp = RB_LEFT(tmp, field); \
        else \
            tmp = RB_RIGHT(tmp, field); \
    } \
    return (parent); \
}

#define RB_NEGINF -1
#define RB_INF 1

#define RB_INSERT(name, x, y) name##_RB_INSERT(x, y)
#define RB_REMOVE(name, x, y) name##_RB_REMOVE(x, y)
#define RB_FIND(name, x, y) name##_RB_FIND(x, y)
#define RB_NFIND(name, x, y) name##_RB_NFIND(x,
y)
#define RB_NEXT(name, x, y) name##_RB_NEXT(y)
#define RB_PREV(name, x, y) name##_RB_PREV(y)
#define RB_MIN(name, x) name##_RB_MINMAX(x, RB_NEGINF)
#define RB_MAX(name, x) name##_RB_MINMAX(x, RB_INF)

#define RB_FOREACH(x, name, head) \
    for ((x) = RB_MIN(name, head); \
        (x) != NULL; \
        (x) = name##_RB_NEXT(x))

#define RB_FOREACH_FROM(x, name, y) \
    for ((x) = (y); \
        ((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
        (x) = (y))

#define RB_FOREACH_SAFE(x, name, head, y) \
    for ((x) = RB_MIN(name, head); \
        ((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
        (x) = (y))

#define RB_FOREACH_REVERSE(x, name, head) \
    for ((x) = RB_MAX(name, head); \
        (x) != NULL; \
        (x) = name##_RB_PREV(x))

#define RB_FOREACH_REVERSE_FROM(x, name, y) \
    for ((x) = (y); \

```

```
((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
(x) = (y))
```

```
#define RB_FOREACH_REVERSE_SAFE(x, name,
head, y) \
for ((x) = RB_MAX(name, head); \
((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
(x) = (y))
```

```
#endif /* _SYS_TREE_H_ */
```

This library is free software; you can redistribute it and/or
modify it under the terms of the Modified BSD License.

The complete text of the license is available in the
../Documentation/licenses/COPYING.BSD-3-Clause file.
BSD 2-Clause License

Copyright (c) 2023, Thorsten Kukuk

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES;

LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later
version.

The complete text of the license is available in the
../Documentation/licenses/COPYING.LGPL-2.1-or-later

```

/*
 * No copyright is claimed. This code is in the public domain; do with
 * it what you wish.
 */
#ifndef UTIL_LINUX_PT_BSD_H
#define UTIL_LINUX_PT_BSD_H

#define BSD_MAXPARTITIONS 16
#define BSD_FS_UNUSED 0

#ifndef BSD_DISKMAGIC
# define BSD_DISKMAGIC ((uint32_t) 0x82564557)
#endif

#define BSD_LINUX_BOOTDIR "/usr/ucb/mdec"

#if defined (__alpha__) || defined (__powerpc__) || \
    defined (__ia64__) || defined (__hppa__)
# define BSD_LABELSECTOR 0
# define BSD_LABELOFFSET 64
#else
# define BSD_LABELSECTOR 1
# define BSD_LABELOFFSET 0
#endif

#define BSD_BBSIZE      8192 /* size of boot area, with label */
#define BSD_SBSIZE      8192 /* max size of fs superblock */

struct bsd_disklabel {
    uint32_t d_magic; /* the magic number */
    int16_t d_type; /* drive type */
    int16_t d_subtype; /* controller/d_type specific */
    char d_typename[16]; /* type name, e.g. "eagle" */
    char d_packname[16]; /* pack identifier */

    /* disk geometry: */
    uint32_t d_secsize; /*
     # of bytes per sector */
    uint32_t d_nsectors; /* # of data sectors per track */
    uint32_t d_ntracks; /* # of tracks per cylinder */
    uint32_t d_ncylinders; /* # of data cylinders per unit */
    uint32_t d_secpercyl; /* # of data sectors per cylinder */
    uint32_t d_secperunit; /* # of data sectors per unit */

    /*
     * Spares (bad sector replacements) below
     * are not counted in d_nsectors or d_secpercyl.
     * Spare sectors are assumed to be physical sectors

```

```

* which occupy space at the end of each track and/or cylinder.
*/
uint16_t d_sparespertrack; /* # of spare sectors per track */
uint16_t d_sparespercyl; /* # of spare sectors per cylinder */

/*
* Alternate cylinders include maintenance, replacement,
* configuration description areas, etc.
*/
uint32_t d_acylinders; /* # of alt. cylinders per unit */

/* hardware characteristics: */
/*
* d_interleave, d_trackskew and d_cylskew describe perturbations
* in the media format used to compensate for a slow
controller.
* Interleave is physical sector interleave, set up by the formatter
* or controller when formatting. When interleaving is in use,
* logically adjacent sectors are not physically contiguous,
* but instead are separated by some number of sectors.
* It is specified as the ratio of physical sectors traversed
* per logical sector. Thus an interleave of 1:1 implies contiguous
* layout, while 2:1 implies that logical sector 0 is separated
* by one sector from logical sector 1.
* d_trackskew is the offset of sector 0 on track N
* relative to sector 0 on track N-1 on the same cylinder.
* Finally, d_cylskew is the offset of sector 0 on cylinder N
* relative to sector 0 on cylinder N-1.
*/
uint16_t d_rpm; /* rotational speed */
uint16_t d_interleave; /* hardware sector interleave */
uint16_t d_trackskew; /* sector 0 skew, per track */
uint16_t d_cylskew; /* sector 0 skew, per cylinder */
uint32_t d_headswitch; /* head switch time, usec */
uint32_t d_trkseek; /*
track-to-track seek, usec */
uint32_t d_flags; /* generic flags */
uint32_t d_drivedata[5]; /* drive-type specific information */
uint32_t d_spare[5]; /* reserved for future use */
uint32_t d_magic2; /* the magic number (again) */
uint16_t d_checksum; /* xor of data incl. partitions */

/* filesystem and partition information: */
uint16_t d_npartitions; /* number of partitions in following */
uint32_t d_bbsize; /* size of boot area at sn0, bytes */
uint32_t d_sbsize; /* max size of fs superblock, bytes */

struct bsd_partition { /* the partition table */

```



```

uint32_t p_size;      /* number of sectors in partition */
uint32_t p_offset;    /* starting sector */
uint32_t p_fsize;     /* filesystem basic fragment size */
uint8_t p_fstype; /* filesystem type, see below */
uint8_t p_frag; /* filesystem fragments per block */
uint16_t p_cpg;      /* filesystem cylinders per group */
} __attribute__((packed))
d_partitions[BSD_MAXPARTITIONS]; /* actually may be more */
} __attribute__((packed));

/* d_type values: */
#define BSD_DTYPE_SMD 1 /* SMD, XSMD; VAX hp/up */
#define BSD_DTYPE_MSCP 2 /* MSCP */
#define BSD_DTYPE_DEC 3 /* other DEC (rk, rl) */
#define BSD_DTYPE_SCSI 4 /* SCSI */
#define BSD_DTYPE_ESDI 5 /* ESDI interface */
#define BSD_DTYPE_ST506 6 /* ST506 etc. */
#define BSD_DTYPE_HPIB 7 /* CS/80 on HP-IB */
#define BSD_DTYPE_HPFL 8 /* HP Fiber-link */
#define BSD_DTYPE_FLOPPY 10 /* floppy */

/* d_subtype values: */
#define BSD_DSTYPE_INDOSPART 0x8 /* is inside dos partition */
#define BSD_DSTYPE_DOSPART(s) ((s) & 3) /* dos partition number */
#define BSD_DSTYPE_GEOMETRY 0x10 /* drive params in label */

/*
 * Filesystem type and version.
 * Used to interpret other filesystem-specific
 * per-partition information.
 */
#define BSD_FS_UNUSED 0 /* unused */
#define BSD_FS_SWAP 1 /* swap */
#define BSD_FS_V6 2 /* Sixth Edition */
#define BSD_FS_V7
    3 /* Seventh Edition */
#define BSD_FS_SYSV 4 /* System V */
#define BSD_FS_V71K 5 /* V7 with 1K blocks (4.1, 2.9) */
#define BSD_FS_V8 6 /* Eighth Edition, 4K blocks */
#define BSD_FS_BSDFFS 7 /* 4.2BSD fast file system */
#define BSD_FS_BSDLFS 9 /* 4.4BSD log-structured file system */
#define BSD_FS_OTHER 10 /* in use, but unknown/unsupported */
#define BSD_FS_HPFS 11 /* OS/2 high-performance file system */
#define BSD_FS_ISO9660 12 /* ISO-9660 filesystem (cdrom) */
#define BSD_FS_ISOFS BSD_FS_ISO9660
#define BSD_FS_BOOT 13 /* partition contains bootstrap */
#define BSD_FS_ADOS 14 /* AmigaDOS fast file system */

```

```

#define BSD_FS_HFS 15 /* Macintosh HFS */
#define BSD_FS_ADVFS 16 /* Digital Unix AdvFS */

/* this is annoying, but it's also the way it is :( */
#ifdef __alpha__
#define BSD_FS_EXT2 8 /* ext2 file system */
#else
#define BSD_FS_MSDFS 8 /* MS-DOS file system */
#endif

/*
 * flags shared by various drives:
 */
#define BSD_D_REMOVABLE 0x01 /*
removable media */
#define BSD_D_ECC 0x02 /* supports ECC */
#define BSD_D_BADSECT 0x04 /* supports bad sector forw. */
#define BSD_D_RAMDISK 0x08 /* disk emulator */
#define BSD_D_CHAIN 0x10 /* can do back-back transfers */
#define BSD_D_DOSPART 0x20 /* within MSDOS partition */

#endif /* UTIL_LINUX_PT_BSD_H */

```

1.6 libxml2 2.13.9-r0

1.6.1 Available under license :

Except where otherwise noted in the source code (e.g. the files dict.c and list.c, which are covered by a similar licence but with different Copyright notices) all the files are:

Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.7 idna 15.1.0

1.7.1 Available under license :

BSD-style without advertising clause

1.8 libbsd 0.12.2-r0

1.8.1 Available under license :

Format: <https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Files:

*

Copyright:

Copyright 2004-2024 Guillem Jover <guillem@hadrons.org>

License: BSD-3-clause

Files:

include/bsd/err.h

include/bsd/stdlib.h

include/bsd/sys/param.h

include/bsd/unistd.h

src/bsd_getopt.c

src/err.c

src/fgetln.c

src/progname.c

Copyright:

Copyright 2005, 2008-2012, 2019 Guillem Jover <guillem@hadrons.org>

Copyright 2005 Hector Garcia Alvarez

Copyright 2005 Aurelien Jarno

Copyright 2006 Robert Millan

Copyright 2018 Facebook, Inc.

License: BSD-3-clause

Files:

include/bsd/netinet/ip_icmp.h

include/bsd/sys/bitstring.h

include/bsd/sys/queue.h

include/bsd/sys/time.h

include/bsd/timeconv.h

include/bsd/vis.h

man/bitstring.3bsd

man/errc.3bsd

man/explicit_bzero.3bsd

man/fgetln.3bsd

man/fgetwln.3bsd

man/fpurge.3bsd

man/funopen.3bsd

man/getbsize.3bsd

man/heapsort.3bsd

man/nlist.3bsd

man/pwcache.3bsd

man/queue.3bsd

man/radixsort.3bsd

man/reallocarray.3bsd

man/reallocf.3bsd

man/setmode.3bsd

man/strmode.3bsd

man/strnstr.3bsd

man/strtoi.3bsd

man/strtou.3bsd

man/unvis.3bsd

man/vis.3bsd

man/wcsncpy.3bsd

src/getbsize.c

src/heapsort.c

src/merge.c

src/nlist.c

src/pwcache.c

src/radixsort.c

src/setmode.c

src/strmode.c

src/strnstr.c

src/strtoi.c

src/strtou.c

src/unvis.c

Copyright:

Copyright 1980, 1982, 1986, 1989-1994

The Regents of the University of California. All rights reserved.

Copyright 1992 Keith Muller.

Copyright 2001 Mike Barcroft <mike@FreeBSD.org>

.

Some code is derived from software contributed to Berkeley by the American National Standards Committee X3, on Information Processing Systems.

.

Some code is derived from software contributed to Berkeley by Peter McIlroy.

.

Some code is derived from software contributed to Berkeley by Ronnie Kon at Mindcraft Inc., Kevin Lew and Elmer Yglesias.

.
Some code is derived from software contributed to Berkeley by
Dave Borman at Cray Research, Inc.

.
Some code is
derived from software contributed to Berkeley by
Paul Vixie.

.
Some code is derived from software contributed to Berkeley by
Chris Torek.

.
Copyright UNIX System Laboratories, Inc.
All or some portions of this file are derived from material licensed
to the University of California by American Telephone and Telegraph
Co. or Unix System Laboratories, Inc. and are reproduced herein with
the permission of UNIX System Laboratories, Inc.
License: BSD-3-clause-Regents

Files:
src/vis.c
Copyright:
Copyright 1989, 1993
The Regents of the University of California. All rights reserved.

.
Copyright 1999, 2005 The NetBSD Foundation, Inc.
All rights reserved.
License: BSD-3-clause-Regents and BSD-2-clause-NetBSD

Files:
include/bsd/libutil.h
Copyright:
Copyright 1996 Peter Wemm <peter@FreeBSD.org>.
All rights reserved.
Copyright 2002 Networks Associates Technology, Inc.
All rights reserved.
License: BSD-3-clause-author

Files:
man/timeradd.3bsd
Copyright:
Copyright
2009 Jukka Ruohonen <jruohonen@iki.fi>
Copyright 1999 Kelly Yancey <kbyanc@posi.net>
All rights reserved.
License: BSD-3-clause-John-Birrell
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the author nor the names of any co-contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY JOHN BIRRELL AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:

man/setproctitle.3bsd

Copyright:

Copyright 1995 Peter Wemm <peter@FreeBSD.org>

All rights reserved.

License: BSD-5-clause-Peter-Wemm

Redistribution and use in source and binary forms, with or without modification, is permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice immediately at the beginning of the file, without modification, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. This work was done expressly for inclusion into FreeBSD. Other use is permitted provided this notation is included.
4. Absolutely no warranty of function or purpose is made by the author Peter Wemm.
5. Modifications may be freely made to this file providing the above conditions are met.

Files:

include/bsd/stringlist.h

man/arc4random.3bsd

man/fmtcheck.3bsd
man/humanize_number.3bsd
man/stringlist.3bsd
man/timeval.3bsd
src/fmtcheck.c
src/humanize_number.c
src/stringlist.c
src/strtonum.c

Copyright:

Copyright 1994, 1997-2000, 2002, 2008, 2010, 2014

The NetBSD Foundation, Inc.

Copyright 2013 John-Mark Gurney <jmg@FreeBSD.org>

All rights reserved.

.

Copyright 2014

The NetBSD Foundation, Inc.

All rights reserved.

.

Some code was derived from software contributed to The NetBSD Foundation by Taylor R. Campbell.

.

Some code was contributed to The NetBSD Foundation by Allen Briggs.

.

Some code was contributed to The NetBSD Foundation by Luke Mewburn.

.

Some code is derived from software contributed to The NetBSD Foundation by Jason R. Thorpe of the Numerical Aerospace Simulation Facility, NASA Ames Research Center, by Luke Mewburn and by Tomas Svensson.

.

Some code is derived from software contributed to The NetBSD Foundation by Julio M. Merino Vidal, developed as part of Google's Summer of Code 2005 program.

.

Some code is derived from software contributed to The NetBSD Foundation by Christos Zoulas.

.

Some code is derived from software contributed to The NetBSD Foundation by Jukka Ruohonen.

License: BSD-2-clause-NetBSD

Files:

include/bsd/sys/endian.h
man/byteorder.3bsd
man/closefrom.3bsd
man/expand_number.3bsd
man/flopen.3bsd
man/getpeereid.3bsd
man/pidfile.3bsd

src/expand_number.c

src/pidfile.c

src/reallocf.c

src/timeconv.c

Copyright:

Copyright 1998, M. Warner Losh <imp@freebsd.org>

All rights reserved.

.

Copyright 2001 Dima Dorfman.

All rights reserved.

.

Copyright 2001 FreeBSD Inc.

All rights reserved.

.

Copyright 2002 Thomas Moestl <tmml@FreeBSD.org>

All rights reserved.

.

Copyright 2002 Mike Barcroft <mike@FreeBSD.org>

All rights reserved.

.

Copyright 2005 Pawel Jakub Dawidek <pjd@FreeBSD.org>

All rights reserved.

.

Copyright 2005 Colin Percival

All rights reserved.

.

Copyright 2007 Eric Anderson <anderson@FreeBSD.org>

Copyright 2007 Pawel Jakub Dawidek <pjd@FreeBSD.org>

All rights reserved.

.

Copyright 2007 Dag-Erling Codan Smrgrav

All rights reserved.

.

Copyright 2009 Advanced Computing Technologies LLC

All rights reserved.

.

Copyright 2011 Guillem Jover <guillem@hadrons.org>

License: BSD-2-clause

Files:

src/flopen.c

Copyright:

Copyright 2007-2009

Dag-Erling Codan Smrgrav

All rights reserved.

License: BSD-2-clause-verbatim

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions

are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer in this position and unchanged.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:

include/bsd/sys/tree.h
man/fparseln.3bsd
man/tree.3bsd
src/fparseln.c

Copyright:

Copyright 1997 Christos Zoulas.
All rights reserved.

Copyright 2002 Niels Provos <provos@citi.umich.edu>
All rights reserved.
License: BSD-2-clause-author

Files:

include/bsd/readpassphrase.h
man/readpassphrase.3bsd
man/strncpy.3bsd
man/strtonum.3bsd
src/arc4random.c
src/arc4random_linux.h
src/arc4random_uniform.c
src/arc4random_unix.h
src/arc4random_win.h
src/closefrom.c
src/freezero.c
src/getentropy_aix.c
src/getentropy_bsd.c

src/getentropy_hpux.c
src/getentropy_hurd.c
src/getentropy_linux.c
src/getentropy_osx.c
src/getentropy_solaris.c
src/getentropy_win.c
src/readpassphrase.c
src/reallocarray.c
src/recallocarray.c
src/strlcat.c
src/strncpy.c
test/explicit_bzero.c
test/strtonum.c

Copyright:

Copyright 2004 Ted Unangst and Todd Miller

All rights reserved.

.

Copyright 1996 David Mazieres <dm@uun.org>

Copyright 1998, 2000-2002, 2004-2005, 2007, 2010, 2012-2015

Todd C. Miller <Todd.Miller@courtesan.com>

Copyright 2004 Ted Unangst

Copyright 2004 Otto Moerbeek <otto@drijf.net>

Copyright 2008 Damien Miller <djm@openbsd.org>

Copyright 2008, 2010-2011, 2016-2017 Otto Moerbeek <otto@drijf.net>

Copyright 2013 Markus Friedl <markus@openbsd.org>

Copyright 2014 Bob Beck <beck@obtuse.com>

Copyright 2014 Brent Cook <bcook@openbsd.org>

Copyright 2014 Pawel Jakub Dawidek <pjd@FreeBSD.org>

Copyright 2014 Theo de Raadt <deraadt@openbsd.org>

Copyright 2014 Google Inc.

Copyright 2015 Michael Felt <aixtools@gmail.com>

Copyright 2015, 2022 Guillem Jover <guillem@hadrons.org>

License:

ISC

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Files:

src/inet_net_pton.c

Copyright:

Copyright 1996 by Internet Software Consortium.

License: ISC-Original

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

.

THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Files:

src/setproctitle.c

Copyright:

Copyright 2010 William Ahern

Copyright 2012 Guillem Jover <guillem@hadrons.org>

License: Expat

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Files:

src/explicit_bzero.c

src/chacha_private.h

Copyright:

None

License: public-domain

Public domain.

Files:

man/mdX.3bsd

Copyright:

None

License: Beerware

"THE BEER-WARE LICENSE" (Revision 42):

<phk@login.dkuug.dk> wrote this

file. As long as you retain this notice you

can do whatever you want with this stuff. If we meet some day, and you think this stuff is worth it, you can buy me a beer in return. Poul-Henning Kamp

License: BSD-3-clause-Regents

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-3-clause-author

Redistribution and use in source and binary forms, with or without modification, is permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the

documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-3-clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-2-clause-NetBSD

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-2-clause-author

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-2-clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.9 python-requests 2.32.3

1.9.1 Available under license :

Requests

Copyright 2019 Kenneth Reitz

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,

including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of

this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

1.10 packaging 25.0

1.10.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

This software is made available under the terms of **either** of the licenses found in LICENSE.APACHE or LICENSE.BSD. Contributions to this software is made under the terms of **both** these licenses.

Copyright (c) Donald Stufft and individual contributors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.11 pax-utils 1.3.8-r1

1.11.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any

patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish,

that in

whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can

be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program

(or a work based on it,

under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your

cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates

the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

1.12 elfutils 0.193-r0

1.12.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by
the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an

Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

- a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or
- b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

- a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the

Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
- b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

Format: <https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Upstream-Name: compressme-seekable-xz

Upstream-Contact: Omar Sandoval <osandov@fb.com>

Files: *

Copyright: Meta Platforms, Inc. and affiliates.

License: GPL-2.0+

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

.

On Debian systems, the complete text of the GNU General Public License

Version 2 can be found in `/usr/share/common-licenses/GPL-2'.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for

this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work

are not derived from the Program,
and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary

form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then

the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free

Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE,
THERE IS NO WARRANTY

FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU

General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute
it
under certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than ``show w'` and ``show c'`; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this
is what you want to do, use the GNU Lesser General
Public License instead of this License.

This license applies to the eu-readelf.1 man page which was forked
from the binutils readelf version of the man page. The rest of the
documentation is provided under the license found in the top level

directory.

GNU Free Documentation License
Version 1.3, 3 November 2008

Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
<<https://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other
functional and useful document "free" in the sense of freedom: to
assure everyone the effective freedom to copy and redistribute it,
with or without modifying it, either commercially or noncommercially.
Secondarily, this License preserves for the author and publisher a way
to get credit for their work, while not being considered responsible
for modifications made by others.

This License
is a kind of "copyleft", which means that derivative
works of the document must themselves be free in the same sense. It
complements the GNU General Public License, which is a copyleft
license designed for free software.

We have designed this License in order to use it for manuals for free
software, because free software needs free documentation: a free
program should come with manuals providing the same freedoms that the
software does. But this License is not limited to software manuals;
it can be used for any textual work, regardless of subject matter or
whether it is published as a printed book. We recommend this License
principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that
contains a notice placed by the copyright holder saying it can be
distributed under the terms of this License. Such a notice grants a
world-wide, royalty-free license, unlimited in duration, to
use that
work under the conditions stated herein. The "Document", below,
refers to any such manual or work. Any member of the public is a
licensee, and is addressed as "you". You accept the license if you
copy, modify or distribute the work in a way requiring permission
under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of

transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and

you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition.

Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material.

If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an

Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release

the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document,

unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the

list of Invariant Sections in the Modified Version's license notice.

These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and

multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the

electronic equivalent of covers if the Document is in electronic form.

Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4.

Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after

your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <https://www.gnu.org/licenses/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0

license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in

part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the

License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies

of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer

can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand

ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The

"System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically

linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered

by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed;

section 10

makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention

is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately
publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to
"keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to

copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install

modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a)

provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have

actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17.

Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

1.13 valgrind 3.25.1-r1

1.13.1 Available under license :

```
head 1.1;
branch 1.1.1;
access;
symbols
VALGRIND_2_0_REALLY:1.1.1.1.0.12
VALGRIND_20030725:1.1.1.1
VALGRIND_20030716:1.1.1.1
VALGRIND_1_9_5:1.1.1.1
VALGRIND_2_0_BRANCH:1.1.1.1.0.10
VALGRIND_1_9_4:1.1.1.1
HEAD_post_dir_restructure:1.1.1.1
VALGRIND_1_0_3:1.1.1.1
```

VALGRIND_1_0_2:1.1.1.1
VALGRIND_1_0_1:1.1.1.1
ERASER:1.1.1.1.0.8
VALGRIND_1_0_0:1.1.1.1
VALGRIND_1_0_BRANCH:1.1.1.1.0.6
DIDUCE:1.1.1.1.0.4
NEWMEM:1.1.1.1.0.2
BEFORE_CACHEGRIND:1.1.1.1
start:1.1.1.1
vendor:1.1.1;
locks; strict;
comment @# @;

1.1
date 2002.03.22.01.27.57; author sewardj; state Exp;
branches
1.1.1.1;
next ;

1.1.1.1
date 2002.03.22.01.27.57; author sewardj; state Exp;
branches;
next ;

desc
@@

1.1
log
@Initial revision
@
text
@ GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document,
but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your

freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law; that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively

when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License.

However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions

on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will

be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this

License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest

possible use to the public,
the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public
License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your
employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.

@

1.1.1.1

log

@Import (snapshot 20020317 + Frederic Gobry's autoconf work).

@

text

@@

1.14 expat 2.7.3-r0

1.14.1 Available under license :

Copyright (c) 1998-2000 Thai Open Source Software Center Ltd and Clark Cooper
Copyright (c) 2001-2025 Expat maintainers

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.15 more-itertools 10.3.0

1.15.1 Available under license :

Copyright (c) 2012 Erik Rose

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.16 krb5-conf 1.0-r2

1.16.1 Available under license :

[NOTE: MIT has only incorporated the mechglue and spnego change, and not the incremental propagation changes. The filenames are different between the Sun and MIT sources. The actual MIT filenames appear in the top-level README file. Original text of Sun's LICENSE file follows.]

Subject to the license set forth below, Sun Microsystems, Inc. donates the attached files to MIT for the purpose of including these modifications and additions in future versions of the Kerberos system.

Many of the files attached are subject to licenses issued by other entities, including OpenVision, MIT, and FundsXpress. See the individual files, and/or related Readme files, for these licenses.

In addition Sun requires that the license set forth below be incorporated into any future version of the Kerberos system which contains portions of the files attached. The following files must be listed, in the top level Readme file, as being provided subject to such

license:

cmd/krb5/ipropl/ipropl.x
cmd/krb5/ipropl/ipropl_hdr.h
cmd/krb5/kadmin/server/ipropld_svc.c
cmd/krb5/kproplog/kproplog.c
cmd/krb5/slave/kpropd_rpc.c
lib/gss_mechs/mech_krb5/et/kdb5_err.c
lib/gss_mechs/mech_spnego/mech/gssapiP_spnego.h
lib/gss_mechs/mech_spnego/mech/spnego_mech.c
lib/krb5/kadm5/kadm_host_srv_names.c
lib/krb5/kdb/kdb_convert.c
lib/krb5/kdb/kdb_hdr.h
lib/krb5/kdb/kdb_log.c
lib/krb5/kdb/kdb_log.h
lib/libgss/g_accept_sec_context.c
lib/libgss/g_acquire_cred.c
lib/libgss/g_canon_name.c
lib/libgss/g_compare_name.c
lib/libgss/g_context_time.c
lib/libgss/g_delete_sec_context.c
lib/libgss/g_dsp_name.c
lib/libgss/g_dsp_status.c
lib/libgss/g_dup_name.c
lib/libgss/g_exp_sec_context.c
lib/libgss/g_export_name.c
lib/libgss/g_glue.c
lib/libgss/g_imp_name.c
lib/libgss/g_imp_sec_context.c
lib/libgss/g_init_sec_context.c
lib/libgss/g_initialize.c
lib/libgss/g_inquire_context.c
lib/libgss/g_inquire_cred.c
lib/libgss/g_inquire_names.c
lib/libgss/g_process_context.c
lib/libgss/g_rel_buffer.c
lib/libgss/g_rel_cred.c
lib/libgss/g_rel_name.c
lib/libgss/g_rel_oid_set.c
lib/libgss/g_seal.c
lib/libgss/g_sign.c
lib/libgss/g_store_cred.c
lib/libgss/g_unseal.c
lib/libgss/g_userok.c
lib/libgss/g_utils.c
lib/libgss/g_verify.c
lib/libgss/gssd_pname_to_uid.c
uts/common/gssapi/include/gssapi_err_generic.h

uts/common/gssapi/include/mechglueP.h

Sun's

License is as follows:

Copyright (c) 2004 Sun Microsystems, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright

=====

Copyright |copy| 1985-2020 by the Massachusetts Institute of Technology and its contributors. All rights reserved.

See :ref:`mitK5license` for additional copyright and license information.

.. _mitK5license:

MIT Kerberos License information

=====

.. toctree::

:hidden:

copyright.rst

.. include:: notice.rst

Copyright (C) 1985-2020 by the Massachusetts Institute of Technology.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Downloading of this software may constitute an export of cryptographic software from the United States of America that is subject to the United States Export Administration Regulations (EAR), 15 CFR 730-774. Additional laws or regulations may apply. It is the responsibility of the person or entity contemplating export to comply with all applicable export laws and regulations, including obtaining any required license from the U.S. government.

The U.S. government prohibits export of encryption source code to certain countries and individuals, including, but not limited to, the countries of Cuba, Iran, North Korea, Sudan, Syria, and residents and nationals of those countries.

Documentation components of this software distribution are licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License. (<https://creativecommons.org/licenses/by-sa/3.0/>)

Individual source code files are copyright MIT, Cygnus Support, Novell, OpenVision Technologies, Oracle, Red Hat, Sun Microsystems, FundsXpress, and others.

Project Athena, Athena, Athena MUSE, Discuss, Hesiod, Kerberos, Moira,

and Zephyr are trademarks of the Massachusetts Institute of Technology (MIT). No commercial use of these trademarks may be made without prior written permission of MIT.

"Commercial use" means use of a name in a product or other for-profit manner. It does NOT prevent a commercial firm from referring to the MIT trademarks in order to convey information (although in doing so, recognition of their trademark status should be given).

=====

The following copyright and permission notice applies to the
OpenVision Kerberos Administration system located in "kadmin/create",
"kadmin/dbutil", "kadmin/passwd", "kadmin/server", "lib/kadm5", and
portions of "lib/rpc":

Copyright, OpenVision Technologies, Inc., 1993-1996, All Rights
Reserved

WARNING: Retrieving the OpenVision Kerberos Administration system
source code, as described below, indicates your acceptance of the
following terms. If you do not agree to the following terms, do
not retrieve the OpenVision Kerberos administration system.

You may freely use and distribute the Source Code and Object Code
compiled from it, with or without modification, but this Source
Code is provided to you "AS IS" EXCLUSIVE OF ANY WARRANTY,
INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY OR
FITNESS FOR A PARTICULAR PURPOSE, OR ANY OTHER WARRANTY, WHETHER
EXPRESS OR IMPLIED. IN NO EVENT WILL OPENVISION HAVE ANY LIABILITY
FOR ANY LOST PROFITS, LOSS OF DATA OR COSTS OF PROCUREMENT OF
SUBSTITUTE
GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, INCLUDING,
WITHOUT LIMITATION, THOSE RESULTING FROM THE USE OF THE SOURCE
CODE, OR THE FAILURE OF THE SOURCE CODE TO PERFORM, OR FOR ANY
OTHER REASON.

OpenVision retains all copyrights in the donated Source Code.
OpenVision also retains copyright to derivative works of the Source
Code, whether created by OpenVision or by a third party. The
OpenVision copyright notice must be preserved if derivative works
are made based on the donated Source Code.

OpenVision Technologies, Inc. has donated this Kerberos
Administration system to MIT for inclusion in the standard Kerberos
5 distribution. This donation underscores our commitment to

continuing Kerberos technology development and our gratitude for the valuable work which has been performed by MIT and the Kerberos community.

=====

Portions contributed by Matt Crawford "crawd@fnal.gov" were work performed at Fermi National Accelerator Laboratory, which is operated by Universities Research Association, Inc., under contract DE-AC02-76CHO3000 with the U.S. Department of Energy.

=====

Portions of "src/lib/crypto" have the following copyright:

Copyright (C) 1998 by the FundsXpress, INC.

All rights reserved.

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of FundsXpress. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. FundsXpress makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

=====

The implementation of the AES encryption algorithm in "src/lib/crypto/builtin/aes" has the following copyright:

Copyright (C) 2001, Dr Brian Gladman "brg@gladman.uk.net", Worcester, UK.
All rights reserved.

LICENSE TERMS

The free distribution and use of this software in both source and binary form is allowed (with or without changes) provided that:

1. distributions of this source code include the above copyright notice, this list of conditions and the following disclaimer;
2. distributions in binary form include the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other associated materials;
3. the copyright holder's name is not used to endorse products built using this software without specific written permission.

DISCLAIMER

This software is provided 'as is' with no explicit or implied warranties in respect of any properties, including, but not limited to, correctness and fitness for purpose.

=====

Portions contributed by Red Hat, including the pre-authentication plug-in framework and the NSS crypto implementation, contain the following copyright:

Copyright (C) 2006 Red Hat, Inc.
Portions
copyright (C) 2006 Massachusetts Institute of Technology
All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Red Hat, Inc., nor the names of its contributors may be used to endorse or promote products derived

from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS

FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

The bundled verto source code is subject to the following license:

Copyright 2011 Red Hat, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

The MS-KKDCP client implementation has the following copyright:

Copyright 2013,2014

Red Hat, Inc.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

The implementations of GSSAPI mechglue in GSSAPI-SPNEGO in "src/lib/gssapi", including the following files:

lib/gssapi/generic/gssapi_err_generic.et
lib/gssapi/mechglue/g_accept_sec_context.c
lib/gssapi/mechglue/g_acquire_cred.c
lib/gssapi/mechglue/g_canon_name.c
lib/gssapi/mechglue/g_compare_name.c
lib/gssapi/mechglue/g_context_time.c
lib/gssapi/mechglue/g_delete_sec_context.c
lib/gssapi/mechglue/g_dsp_name.c
lib/gssapi/mechglue/g_dsp_status.c
lib/gssapi/mechglue/g_dup_name.c
lib/gssapi/mechglue/g_exp_sec_context.c
lib/gssapi/mechglue/g_export_name.c
lib/gssapi/mechglue/g_glue.c
lib/gssapi/mechglue/g_imp_name.c

lib/gssapi/mechglue/g_imp_sec_context.c
lib/gssapi/mechglue/g_init_sec_context.c
lib/gssapi/mechglue/g_initialize.c
lib/gssapi/mechglue/g_inquire_context.c
lib/gssapi/mechglue/g_inquire_cred.c
lib/gssapi/mechglue/g_inquire_names.c
lib/gssapi/mechglue/g_process_context.c
lib/gssapi/mechglue/g_rel_buffer.c
lib/gssapi/mechglue/g_rel_cred.c
lib/gssapi/mechglue/g_rel_name.c
lib/gssapi/mechglue/g_rel_oid_set.c
lib/gssapi/mechglue/g_seal.c
lib/gssapi/mechglue/g_sign.c
lib/gssapi/mechglue/g_store_cred.c
lib/gssapi/mechglue/g_unseal.c
lib/gssapi/mechglue/g_userok.c
lib/gssapi/mechglue/g_utils.c
lib/gssapi/mechglue/g_verify.c
lib/gssapi/mechglue/gssd_pname_to_uid.c
lib/gssapi/mechglue/mglueP.h
lib/gssapi/mechglue/oid_ops.c
lib/gssapi/spnego/gssapiP_spnego.h
lib/gssapi/spnego/spnego_mech.c

and the initial implementation of incremental propagation,
including
the following new or changed files:

include/iprop_hdr.h
kadmin/server/ipropd_svc.c
lib/kdb/iprop.x
lib/kdb/kdb_convert.c
lib/kdb/kdb_log.c
lib/kdb/kdb_log.h
lib/krb5/error_tables/kdb5_err.et
kprop/kpropd_rpc.c
kprop/kproplog.c

are subject to the following license:

Copyright (C) 2004 Sun Microsystems, Inc.

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use, copy,
modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

Kerberos V5 includes documentation and software developed at the University of California at Berkeley, which includes this copyright notice:

Copyright (C) 1983 Regents of the University of California.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====
Portions contributed by Novell, Inc., including the LDAP database backend, are subject to the following license:

Copyright (C) 2004-2005, Novell, Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * The copyright holder's name is not used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions

funded by Sandia National Laboratory and developed by the University of Michigan's Center for Information Technology Integration, including the PKINIT implementation, are subject to the following license:

COPYRIGHT (C) 2006-2007
THE REGENTS OF THE UNIVERSITY OF MICHIGAN
ALL RIGHTS RESERVED

Permission is granted to use, copy, create derivative works and redistribute this software and such derivative works for any purpose, so long as the name of The University of Michigan is not used in any advertising or publicity pertaining to the use of distribution of this software without specific, written prior authorization. If the above copyright notice or any other identification of the University of Michigan is included in any copy of any portion of this software, then the disclaimer below must also be included.

THIS SOFTWARE IS PROVIDED AS IS, WITHOUT REPRESENTATION FROM THE UNIVERSITY OF MICHIGAN
AS TO ITS FITNESS FOR ANY PURPOSE, AND
WITHOUT WARRANTY BY THE UNIVERSITY OF MICHIGAN OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE REGENTS OF THE UNIVERSITY OF MICHIGAN SHALL NOT BE LIABLE FOR ANY DAMAGES, INCLUDING SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WITH RESPECT TO ANY CLAIM ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE SOFTWARE, EVEN IF IT HAS BEEN OR IS HEREAFTER ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

=====

The pkcs11.h file included in the PKINIT code has the following license:

Copyright 2006 g10 Code GmbH
Copyright 2006 Andreas Jellinghaus

This file is free software; as a special exception the author gives unlimited permission to copy and/or distribute it, with or without modifications, as long as this notice is preserved.

This file is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY, to the extent permitted by law; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR

PURPOSE.

=====

Portions contributed by Apple Inc. are subject to the following license:

Copyright 2004-2008 Apple Inc. All Rights Reserved.

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Apple Inc. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Apple Inc. makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

=====

The implementations of UTF-8 string handling in src/util/support and src/lib/krb5/unicode are subject to the following copyright and permission notice:

The OpenLDAP Public License
Version 2.8, 17 August 2003

Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that the following conditions are met:

1. Redistributions in source form must retain copyright statements and notices,
2. Redistributions in binary form must reproduce applicable

copyright statements and notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution, and

3. Redistributions must contain a verbatim copy of this document.

The OpenLDAP Foundation may revise this license from time to time. Each revision is distinguished by a version number. You may use this Software under terms of this license revision or under the terms of any subsequent revision of the license.

THIS SOFTWARE IS PROVIDED BY THE OPENLDAP FOUNDATION AND ITS CONTRIBUTORS "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OPENLDAP FOUNDATION, ITS CONTRIBUTORS, OR THE AUTHOR(S) OR OWNER(S) OF THE SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The names of the authors and copyright holders must not be used in advertising or otherwise to promote the sale, use or other dealing in this Software without specific, written prior permission. Title to copyright in this Software shall at all times remain with copyright holders.

OpenLDAP is a registered trademark of the OpenLDAP Foundation.

Copyright 1999-2003 The OpenLDAP Foundation, Redwood City, California, USA. All Rights Reserved. Permission to copy and distribute verbatim copies of this document is granted.

=====

Marked test programs in src/lib/krb5/krb have the following copyright:

Copyright (C) 2006 Kungliga Tekniska Hgskola
(Royal Institute of Technology, Stockholm, Sweden).
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of KTH nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY KTH AND ITS CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL KTH OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

The KCM Mach RPC definition file used on macOS
has the following
copyright:

Copyright (C) 2009 Kungliga Tekniska Hgskola
(Royal Institute of Technology, Stockholm, Sweden).
All rights reserved.

Portions Copyright (C) 2009 Apple Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE
AND CONTRIBUTORS "AS IS"

AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

Portions of the RPC implementation in src/lib/rpc and
src/include/gssrpc have the following copyright and permission notice:

Copyright (C) 2010, Oracle America, Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the "Oracle America, Inc." nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

Copyright (C) 2006,2007,2009 NTT (Nippon Telegraph and Telephone Corporation). All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer as the first lines of this file unmodified.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY NTT "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL NTT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED

OF THE POSSIBILITY OF SUCH
DAMAGE.

=====

Copyright 2000 by Carnegie Mellon University

All Rights Reserved

Permission to use, copy, modify, and distribute this software and
its documentation for any purpose and without fee is hereby
granted, provided that the above copyright notice appear in all
copies and that both that copyright notice and this permission
notice appear in supporting documentation, and that the name of
Carnegie Mellon University not be used in advertising or publicity
pertaining to distribution of the software without specific,
written prior permission.

CARNEGIE MELLON UNIVERSITY DISCLAIMS ALL WARRANTIES WITH REGARD TO
THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS, IN NO EVENT SHALL CARNEGIE MELLON UNIVERSITY BE LIABLE
FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER
IN
AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING
OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS
SOFTWARE.

=====

Copyright (C) 2002 Naval Research Laboratory (NRL/CCS)

Permission to use, copy, modify and distribute this software and
its documentation is hereby granted, provided that both the
copyright notice and this permission notice appear in all copies of
the software, derivative works or modified versions, and any
portions thereof.

NRL ALLOWS FREE USE OF THIS SOFTWARE IN ITS "AS IS" CONDITION AND
DISCLAIMS ANY LIABILITY OF ANY KIND FOR ANY DAMAGES WHATSOEVER
RESULTING FROM THE USE OF THIS SOFTWARE.

=====

Copyright (C) 1991, 1992, 1994 by Cygnus Support.

Permission to use, copy, modify, and distribute this software and
its documentation for any purpose and without fee is

hereby
granted, provided that the above copyright notice appear in all
copies and that both that copyright notice and this permission
notice appear in supporting documentation. Cygnus Support makes no
representations about the suitability of this software for any
purpose. It is provided "as is" without express or implied
warranty.

=====
Copyright (C) 2006 Secure Endpoints Inc.

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use, copy,
modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

=====
Portions of the implementation of the Fortuna-like PRNG are subject to
the following notice:

Copyright (C) 2005 Marko Kreen
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above
copyright

notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 1994 by the University of Southern California

EXPORT OF THIS SOFTWARE from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to copy, modify, and distribute this software and its documentation in source and binary forms is hereby granted, provided that any documentation or other materials related to such distribution or use acknowledge that the software was developed by the University of Southern California.

DISCLAIMER OF WARRANTY. THIS SOFTWARE IS PROVIDED "AS IS". The University of Southern California MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. By way of example, but not limitation, the University of Southern California MAKES NO REPRESENTATIONS OR WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. The University of Southern California shall not be held liable for any liability nor for any direct, indirect, or consequential damages with respect to any claim by the user or distributor of the ksu software.

=====

This code is derived from software contributed to Harvard by Jeremy Rassen.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

This product includes software developed by the University of California, Berkeley and its contributors.

4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

Copyright (C) 2008 by the Massachusetts Institute of Technology.
Copyright 1995 by Richard P. Basch. All Rights Reserved.

Copyright 1995 by Lehman Brothers, Inc. All Rights Reserved.

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Richard P. Basch, Lehman Brothers and M.I.T. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Richard P. Basch, Lehman Brothers and M.I.T. make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

=====

The following notice applies to "src/lib/krb5/krb/strptime.c" and "src/include/k5-queue.h".

Copyright (C) 1997, 1998 The NetBSD Foundation, Inc.
All rights reserved.

This code was contributed to The NetBSD Foundation by Klaus Klein.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

This product includes software developed by the NetBSD

Foundation, Inc. and its contributors.

4. Neither the name of The NetBSD Foundation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

The following notice applies to Unicode library files in "src/lib/krb5/unicode":

Copyright 1997, 1998, 1999 Computing Research Labs,
New Mexico State University

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE COMPUTING RESEARCH LAB OR NEW MEXICO STATE UNIVERSITY BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR

OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE
OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.

=====

The following notice applies to "src/util/support/strncpy.c":

Copyright (C) 1998 Todd C. Miller "Todd.Miller@courtesan.com"

Permission to use, copy, modify, and distribute this software for
any purpose with or without fee is hereby granted, provided that
the above copyright notice and this permission notice appear in all
copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL
WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE
AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR
CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS
OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN
CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

=====

The
following notice applies to "src/util/profile/argv_parse.c" and
"src/util/profile/argv_parse.h":

Copyright 1999 by Theodore Ts'o.

Permission to use, copy, modify, and distribute this software for
any purpose with or without fee is hereby granted, provided that
the above copyright notice and this permission notice appear in all
copies. THE SOFTWARE IS PROVIDED "AS IS" AND THEODORE TS'O (THE
AUTHOR) DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE,
INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN
NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT,
INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER
RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION
OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR
IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. (Isn't
it sick that the U.S. culture of lawsuit-happy lawyers requires
this
kind of disclaimer?)

=====

The following notice applies to SWIG-generated code in
"src/util/profile/profile_tcl.c":

Copyright (C) 1999-2000, The University of Chicago

This file may be freely redistributed without license or fee
provided this copyright message remains intact.

=====

The following notice applies to portions of "src/lib/rpc" and
"src/include/gssrpc":

Copyright (C) 2000 The Regents of the University of Michigan. All
rights reserved.

Copyright (C) 2000 Dug Song "dugsong@UMICH.EDU". All rights
reserved, all wrongs reversed.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above
copyright notice, this list of conditions and the following
disclaimer.
2. Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials provided
with the distribution.
3. Neither the name of the University nor the names of its
contributors may be used to endorse or promote products derived
from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT
OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT

LIABILITY, OR TORT

(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

Implementations of the MD4 algorithm are subject to the following notice:

Copyright (C) 1990, RSA Data Security, Inc. All rights reserved.

License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD4 Message Digest Algorithm" in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD4 Message Digest Algorithm" in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

=====

Implementations of the MD5 algorithm are subject to the following notice:

Copyright (C) 1990, RSA Data Security, Inc. All rights reserved.

License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

=====

The following notice applies to
"src/lib/crypto/crypto_tests/t_mddriver.c":

Copyright (C) 1990-2, RSA Data Security, Inc. Created 1990. All rights reserved.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

=====

Portions of "src/lib/krb5" are subject to the following notice:

Copyright (C) 1994 CyberSAFE Corporation.
Copyright 1990,1991,2007,2008 by the Massachusetts
Institute of Technology.
All Rights Reserved.

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of M.I.T. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Furthermore if you modify this software

you must label your software as modified software and not distribute it in such a fashion that it might be confused with the original M.I.T. software. Neither M.I.T., the Open Computing Security Group, nor CyberSAFE Corporation make any representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

=====

Portions contributed by PADL Software are subject to the following license:

Copyright (c) 2011, PADL Software Pty Ltd. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following

disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of PADL Software nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY PADL SOFTWARE AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL PADL SOFTWARE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

The bundled libev source code is subject to the following license:

All files in libev are Copyright (C)2007,2008,2009 Marc Alexander Lehmann.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS

FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Alternatively, the contents of this package may be used under the terms of the GNU General Public License ("GPL") version 2 or any later version, in which case the provisions of the GPL are applicable instead of the above. If you wish to allow the use of your version of this package only under the terms of the GPL and not to allow others to use your version of this file under the BSD

license, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the GPL in this and the other files of this package. If you do not delete the provisions above, a recipient may use your version of this file under either the BSD or the GPL.

=====

Files copied from the Intel AESNI Sample Library are subject to the

following license:

Copyright (C) 2010, Intel Corporation All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer

in the documentation and/or other materials provided with the distribution.

- * Neither the name of Intel Corporation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

The following notice applies to
"src/ccapi/common/win/OldCC/autolock.hxx":

Copyright (C) 1998 by Danilo Almeida. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

The following notice applies to portions of
"src/plugins/preauth/spake/edwards25519.c" and
"src/plugins/preauth/spake/edwards25519_tables.h":

The MIT License (MIT)

Copyright (c) 2015-2016 the fiat-crypto authors (see the AUTHORS file).

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

The
following notice applies to portions of
"src/plugins/preauth/spake/edwards25519.c":

Copyright (c) 2015-2016, Google Inc.

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.17 libffi 3.4.8-r0

1.17.1 Available under license :

The libffi source distribution contains certain code that is not part of libffi, and is only used as tooling to assist with the building and testing of libffi. This includes the msvcc.sh script used to wrap the Microsoft compiler with GNU compatible command-line options, make_sunver.pl, and the libffi test code distributed in the testsuite/libffi.bhaible directory. This code is distributed with libffi for the purpose of convenience only, and libffi is in no way derived from this code.

msvcc.sh and testsuite/libffi.bhaible are both distributed under the terms of the GNU GPL version 2, as below.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software
are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it. (Some other Free Software Foundation software is covered by
the GNU Lesser General Public License instead.) You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions
that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have. You must make sure that they, too, receive or can get the
source code. And you must show them these terms so they know their
rights.

We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by
others will not reflect on the original
authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily

used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and

all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

You

are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright

holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER

PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be

mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

libffi - Copyright (c) 1996-2022 Anthony Green, Red Hat, Inc and others.
See source files for details.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the ``Software''), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED ``AS IS'', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.18 py3-pip 25.1.1-r0

1.18.1 Available under license :

UNRECOGNIZED LICENSE; MD5 sum: 45874240865793abc3a77b0bbdf3dc51
UNRECOGNIZED LICENSE; MD5 sum: b0eaf0c960a0f5d744339da436682707

UNRECOGNIZED LICENSE; MD5 sum: 856bd8901cf18d4edc40b86012a95b26
UNRECOGNIZED LICENSE; MD5 sum: d2e8c3bce80da2a2e51cffd44205a7b1
UNRECOGNIZED LICENSE; MD5 sum: 89c981cb43c0c7a860b8a18449e92a38
UNRECOGNIZED LICENSE; MD5 sum: f7cacecf6ff92c07405436643ace4302
PKu5Z\&#license.dist-0.2.dist-info/METADATAUT ggux]MN@s
J*VT\$\$ ~nH-dKp>NXw
'h4xOAZu'OR#diT>?ECesh
%W6Q:'nl|^(?t){u-{Z0XPKa5Zhb'!license.dist-0.2.dist-info/RECORDUT ggux}r03I,HZK
m^2fXZ;wzmUnjWU]52+\Gfmy8sUP-"7DbuhoAUo9sM"TT_ST
,9%="la1?O"+l[0|y:4o\$6}D
HzPs_`1rPK5d#Ahr6s%license.dist-0.2.dist-info/RECORD.jwsUT DPmgux%N@EHq()04@:\*`w].9?\'\$mZK'3'
}g7.hI[zMeH4 TQzI9 +g\v7+L? iOU{RC3;>>p_,7:N-f@]i#
;_*bl-X,sZzoI]4a%JQy+KI(:d=^Su`t5U@d"R7PK
d#A~XY@@@ license.dist-0.2.dist-info/WHEELUT FDPmguxWheel-Version: 0.1
Packager: bdist_wheel
Root-Is-Purelib: true

PK
o@licensedist/ __init__.pyUT PBmguxPKu5Z\&#license.dist-0.2.dist-
info/METADATAUTguxPKa5Zhb'!)license.dist-0.2.dist-info/RECORDUTguxPK5d#Ahr6s%Vlicense.dist-0.2.dist-
info/RECORD.jwsUTDPuxPK
d#A~XY@@@ license.dist-0.2.dist-info/WHEELUTFDPuxPK
o@licensedist/ __init__.pyUTPuxPK
SPDX-License-Identifier: MIT
SPDX-FileCopyrightText: 2021 Taneli Hukkinen
Licensed to PSF under a Contributor Agreement.
Copyright (c) 2020 Will McGugan

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

Python/Jython

MIT License

Copyright (c) 2024-present Stephen Rosen <sirosen0@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

UNRECOGNIZED LICENSE; MD5 sum: 969a9e986521a635141d6b548aea92ad

UNRECOGNIZED LICENSE; MD5 sum: c0f4347d81adc62c6cd62b1b85aa469b

BSD 3-Clause License

Copyright (c) 2013-2024, Kim Davies and contributors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT

OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2006-2022 by the respective authors (see AUTHORS file).
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 2008-2011 INADA Naoki <songofacandy@gmail.com>

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

UNRECOGNIZED LICENSE; MD5 sum: 224a0356792d38f867e73fdd7ee8bc68
argparse is (c) 2006-2009 Steven J. Bethard <steven.bethard@gmail.com>.

The argparse module was contributed to Python as of Python 2.7 and thus
was licensed under the Python license. Same license applies to all files in
the argparse package project.

For details about the Python License, please see doc/Python-License.txt.

History

Before (and including) argparse 1.1, the argparse package was licensed under
Apache License v2.0.

After argparse 1.1, all project files from the argparse project were deleted
due to license compatibility issues between Apache License 2.0 and GNU GPL v2.

The project repository then had a clean start with some files taken from
Python 2.7.1, so definitely all files are under Python License now.

=====

Contributors

=====

- * Alex Grnholm
- * Alice Bevan-McGregor
- * Arfrever Frehtes Taifersar Arahesis
- * Christophe Combelles
- * Daniel Stutzbach
- * Daniel Holth
- * Hanno Schlichting
- * Jannis Leidel
- * Jason R. Coombs
- * Jim Fulton
- * Jonathan Lange
- * Justin Azoff
- * Lennart Regebro

- * Marc Abramowitz
- * Martin von Lwis
- * Noufal Ibrahim
- * Pete Hollobon
- * Philip Jenvey
- * Reinout van Rees
- * Robert Myers
- * Stefan H. Holek
- * Tarek Ziad
- * Toshio Kuratomi

If you think you name is missing, please add it (alpha order by first name)
UNRECOGNIZED LICENSE; MD5 sum: b7f43a4fdd86a40df4c36e1b6afca602
Copyright (c) 2011 Agendaless Consulting and Contributors.
(<http://www.agendaless.com>), All Rights Reserved
MIT License

Copyright (c) 2008-2020 Andrey Petrov and contributors (see CONTRIBUTORS.txt)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====
Contributors
=====

- * Alex Grnholm
- * Alice Bevan-McGregor
- * Arfrever Frehtes Taifersar Arahesis
- * Christophe Combelles
- * Daniel Stutzbach
- * Daniel Holth
- * Dirley Rodrigues

- * Donald Stufft
- * Grigory Petrov
- * Hanno Schlichting
- * Jannis Leidel
- * Jason R. Coombs
- * Jim Fulton
- * Jonathan Lange
- * Justin Azoff
- * Lennart Regebro
- * Marc Abramowitz
- * Martin von Lwis
- * Noufal Ibrahim
- * Pete Hollobon
- * Phillip J. Eby
- * Philip Jenvey
- * Philip Thiem
- * Reinout van Rees
- * Robert Myers
- * Stefan H. Holek
- * Tarek Ziad
- * Toshio Kuratomi

If you think your name is missing, please add it (alpha order by first name)

UNRECOGNIZED LICENSE; MD5 sum: 83ef8a4c430ba194704fb45783887e46

UNRECOGNIZED LICENSE; MD5 sum: aa439617d32fe0ada48c0da904366da4

MIT

UNRECOGNIZED LICENSE; MD5 sum: 279f913b168e268d25ef6188c9e3ffd4

Copyright (c) 2008-present The pip developers (see AUTHORS.txt file)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

Copyright (c) 2008-2013 The pip developers (see AUTHORS.txt file)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to

the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

License for Bundle of CA Root Certificates (pip/cacert.pem)

=====

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301

Copyright (c) 2007 Ian Bicking and Contributors
Copyright (c) 2009 Ian Bicking, The Open Planning Project
Copyright (c) 2011-2013 The virtualenv developers

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,

EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES
OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
UNRECOGNIZED LICENSE; MD5 sum: 4ffcc98b2323777b0f5c002e45250264
The MIT License (MIT)

Copyright (c) 2017 Thomas Kluyver

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR

IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

UNRECOGNIZED LICENSE; MD5 sum: 185fbba8d66d86d1c6cc10d6b557cb3f

UNRECOGNIZED LICENSE; MD5 sum: 3fd6103100722c2ab40319eb942c4c6a

UNRECOGNIZED LICENSE; MD5 sum: a7e9b70ca59c69b9aa662e8ae7bbec00

UNRECOGNIZED LICENSE; MD5 sum: 504 added 18bdcfd3f7606b5a0f88addea3

UNRECOGNIZED LICENSE; MD5 sum: bbfcbab4df1f200b2a9ca494c44800c0

A. HISTORY OF THE SOFTWARE

=====

Python was created in the early 1990s by Guido van Rossum at Stichting
Mathematisch Centrum (CWI, see <https://www.cwi.nl>) in the Netherlands
as a successor of a language called ABC. Guido remains Python's
principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for
National Research Initiatives (CNRI, see <https://www.cnri.reston.va.us>)
in Reston, Virginia where he released several versions of the
software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations, which became Zope Corporation. In 2001, the Python Software Foundation (PSF, see <https://www.python.org/psf/>) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation was a sponsoring member of the PSF.

All

Python releases are Open Source (see <https://opensource.org> for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

Release	Derived from	Year	Owner	GPL-compatible? (1)
0.9.0 thru 1.2		1991-1995	CWI	yes
1.3 thru 1.5.2	1.2	1995-1999	CNRI	yes
1.6	1.5.2	2000	CNRI	no
2.0	1.6	2000	BeOpen.com	no
1.6.1	1.6	2001	CNRI	yes (2)
2.1	2.0+1.6.1	2001	PSF	no
2.0.1	2.0+1.6.1	2001	PSF	yes
2.1.1	2.1+2.0.1	2001	PSF	yes
2.1.2	2.1.1	2002	PSF	yes
2.1.3	2.1.2	2002	PSF	yes
2.2 and above	2.1.1	2001-now		
PSF	yes			

Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

=====

Python software and documentation are licensed under the
Python Software Foundation License Version 2.

Starting with Python 3.8.6,
examples, recipes, and other code in
the documentation are dual licensed under the PSF License Version 2
and the Zero-Clause BSD license.

Some software incorporated into Python is under different licenses.
The licenses are listed with code falling under that license.

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.
2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright
(c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010,
2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023 Python Software Foundation;
All Rights Reserved" are retained in Python alone or in any derivative version prepared by Licensee.
3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.
4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.
5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY

INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS

AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at <http://www.pythonlabs.com/logos.html> may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the internet

using the following URL: <http://hdl.handle.net/1895.22/1013>".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam,
The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its
documentation for any purpose and without fee is hereby granted,
provided that the above copyright notice appear in all copies and that
both that copyright notice and this permission notice appear in
supporting documentation, and that the name of Stichting Mathematisch
Centrum or CWI not be used in advertising or publicity pertaining to
distribution of the software without specific, written prior
permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO
THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE
FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER

RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT
OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

ZERO-CLAUSE BSD LICENSE FOR CODE IN THE PYTHON DOCUMENTATION

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH
REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT,
INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM
LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR
OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE.

This package contains a modified version of ca-bundle.crt:

ca-bundle.crt -- Bundle of CA Root Certificates

This is a bundle of X.509 certificates of public Certificate Authorities
(CA). These were automatically extracted from Mozilla's root certificates
file (certdata.txt). This file can be found in the mozilla source tree:
<https://hg.mozilla.org/mozilla-central/file/tip/security/nss/lib/ckfw/builtins/certdata.txt>
It contains the certificates in PEM format and therefore
can be directly used with curl / libcurl / php_curl, or with

an Apache+mod_ssl webserver for SSL client authentication.

Just configure this file as the SSLCertificateFile.#

***** BEGIN LICENSE BLOCK *****

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

***** END LICENSE BLOCK *****

@(#) \$RCSfile: certdata.txt,v \$ \$Revision: 1.80 \$ \$Date: 2011/11/03 15:11:58 \$
UNRECOGNIZED LICENSE; MD5 sum: ee4b86b7d59c7064abc30c2496ac78b9
UNRECOGNIZED LICENSE; MD5 sum: f148932318e88b83787b4b3a0bdcf70a
UNRECOGNIZED LICENSE; MD5 sum: 351b831d68a447eabcbaa43af18d7b41

This software is made available under the terms of *either* of the licenses found in LICENSE.APACHE or LICENSE.BSD. Contributions to this software is made under the terms of *both* these licenses.

UNRECOGNIZED LICENSE; MD5 sum: b8d0bf09a388e9b6640ef5c7c153d61d
MIT License

Copyright (c) 2021 Taneli Hukkinen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2018, Tzu-ping Chung <uranusjr@gmail.com>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR

ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

UNRECOGNIZED LICENSE; MD5 sum: c0c0185b7475d96d2c346c55c3e53ab7

UNRECOGNIZED LICENSE; MD5 sum: f659b2cbaf5278203e32eb1f5a5f79c7

Pylons Project Contributor Agreement

=====

The submitter agrees by adding his or her name within the section below named "Contributors" and submitting the resulting modified document to the canonical shared repository location for this software project (whether directly, as a user with "direct commit access", or via a "pull request"), he or she is signing a contract electronically. The submitter becomes a Contributor after a) he or she signs this document by adding their name beneath the "Contributors" section below, and b) the resulting document is accepted into the canonical version control repository.

Treatment of Account

Contributor will not allow anyone other than the Contributor to use his or her username or source repository login to submit code to a Pylons Project source repository. Should Contributor become aware of any such use, Contributor will immediately by notifying Agendaless Consulting. Notification must be performed by sending an email to webmaster@agendaless.com. Until such notice is received, Contributor will be presumed to have taken all actions made through Contributor's account. If the Contributor has direct commit access, Agendaless Consulting will have complete control and discretion over capabilities assigned to Contributor's account, and may disable Contributor's account for any reason at any time.

Legal Effect of Contribution

Upon submitting a change or new work to a Pylons Project source Repository (a "Contribution"), you agree to assign, and hereby do assign, a one-half interest of all right, title and interest in and to copyright and other intellectual property rights with respect to your new and original portions of the Contribution to Agendaless Consulting. You and Agendaless Consulting each agree that the other shall be free to exercise any and all exclusive rights in and to the Contribution, without accounting to one another, including without limitation, the right to license the Contribution to others under the Repoze Public License. This agreement shall run with title to the Contribution. Agendaless Consulting does not convey to you any right, title or interest in or to the Program or such portions of the Contribution that

were taken from the Program. Your transmission of a submission to the Pylons Project source Repository and marks of identification concerning the Contribution itself constitute your intent to contribute and your assignment of the work in accordance with the provisions of this Agreement.

License Terms

Code committed to the Pylons Project source repository (Committed Code) must be governed by the Repoze Public License (<http://repoze.org/LICENSE.txt>, aka "the RPL") or another license acceptable to Agendaless Consulting. Until Agendaless Consulting declares in writing an acceptable license other than the RPL, only the RPL shall be used. A list of exceptions is detailed within the "Licensing Exceptions" section of this document, if one exists.

Representations, Warranty, and Indemnification

Contributor represents and warrants that the Committed Code does not violate the rights of any person or entity, and that the Contributor has legal authority to enter into this Agreement and legal authority over Contributed Code. Further, Contributor indemnifies Agendaless Consulting against violations.

Cryptography

Contributor understands that cryptographic code may be subject to government regulations with which Agendaless Consulting and/or entities using Committed Code must comply. Any code which contains any of the items listed below must not be checked-in until Agendaless Consulting staff has been notified and has approved such contribution in writing.

- Cryptographic capabilities or features
- Calls to cryptographic features
- User interface elements which provide context relating to cryptography
- Code which may, under casual inspection, appear to be cryptographic.

Notices

Contributor confirms that any notices required will be included in any Committed Code.

Licensing Exceptions

=====

None.

List of Contributors

=====

The below-signed are contributors to a code repository that is part of the project named "Translationstring". Each below-signed contributor has read, understands and agrees to the terms above in the section within this document entitled "Pylons Project Contributor Agreement" as of the date beside his or her name.

Contributors

- Chris McDonough, 2011/02/16

- Wichert Akkerman, 2012/02/02

UNRECOGNIZED LICENSE; MD5 sum: 3f2777d4b97006c32953b26ef1f7ac43

A. HISTORY OF THE SOFTWARE

=====

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see <http://www.cwi.nl>) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see <http://www.cnri.reston.va.us>) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations (now Zope Corporation, see <http://www.zope.com>). In 2001, the Python Software Foundation (PSF, see <http://www.python.org/psf/>) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation is a sponsoring member of the PSF.

All Python releases are Open Source (see <http://www.opensource.org> for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

Release	Derived from	Year	Owner	GPL- compatible? (1)
0.9.0 thru 1.2		1991-1995	CWI	yes
1.3 thru 1.5.2	1.2	1995-1999	CNRI	yes
1.6	1.5.2	2000	CNRI	no
2.0	1.6	2000	BeOpen.com	no
1.6.1	1.6	2001	CNRI	yes (2)
2.1	2.0+1.6.1	2001	PSF	no
2.0.1	2.0+1.6.1	2001	PSF	yes
2.1.1	2.1+2.0.1	2001	PSF	yes
2.2	2.1.1	2001	PSF	yes
2.1.2	2.1.1	2002	PSF	yes
2.1.3				
2.1.2	2002	PSF	yes	
2.2.1	2.2	2002	PSF	yes
2.2.2	2.2.1	2002	PSF	yes
2.2.3	2.2.2	2003	PSF	yes
2.3	2.2.2	2002-2003	PSF	yes
2.3.1	2.3	2002-2003	PSF	yes
2.3.2	2.3.1	2002-2003	PSF	yes
2.3.3	2.3.2	2002-2003	PSF	yes
2.3.4	2.3.3	2004	PSF	yes
2.3.5	2.3.4	2005	PSF	yes
2.4	2.3	2004	PSF	yes
2.4.1	2.4	2005	PSF	yes
2.4.2	2.4.1	2005	PSF	yes
2.4.3	2.4.2	2006	PSF	yes
2.4.4	2.4.3	2006	PSF	yes
2.5	2.4	2006	PSF	yes
2.5.1	2.5	2007	PSF	yes
2.5.2	2.5.1	2008	PSF	yes
2.5.3	2.5.2	2008	PSF	yes
2.6	2.5	2008	PSF	yes
2.6.1	2.6	2008	PSF	yes
2.6.2	2.6.1	2009	PSF	yes
2.6.3	2.6.2	2009	PSF	yes
2.6.4	2.6.3	2009	PSF	yes
2.6.5	2.6.4	2010	PSF	yes
2.7	2.6	2010	PSF	yes

Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with

other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.
2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010 Python Software Foundation; All Rights Reserved" are retained in Python alone or in any derivative version prepared by Licensee.
3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.
4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.
5. PSF SHALL NOT BE LIABLE TO LICENSEE OR

ANY OTHER USERS OF PYTHON
FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS
A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON,
OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE

SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at <http://www.pythonlabs.com/logos.html> may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License

Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This

Agreement may also be obtained from a proxy server on the Internet using the following URL: <http://hdl.handle.net/1895.22/1013>".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S.

federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam,
The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL

DAMAGES OR ANY DAMAGES

WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You"
(or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations,
or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems
that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental,

or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

UNRECOGNIZED LICENSE; MD5 sum: 1545853f429fe0b615b6e97ded0d7631

A. HISTORY OF THE SOFTWARE

=====

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see <http://www.cwi.nl>) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see <http://www.cnri.reston.va.us>) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations (now Zope Corporation, see <http://www.zope.com>). In 2001, the Python Software Foundation (PSF, see <http://www.python.org/psf/>) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation is a sponsoring member of the PSF.

All Python releases are Open Source (see <http://www.opensource.org> for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

Release	Derived from	Year	Owner	GPL-compatible? (1)
0.9.0 thru 1.2		1991-1995	CWI	yes
1.3 thru 1.5.2	1.2	1995-1999	CNRI	yes
1.6	1.5.2	2000	CNRI	no
2.0	1.6	2000	BeOpen.com	no
1.6.1	1.6	2001	CNRI	yes (2)
2.1	2.0+1.6.1	2001	PSF	no
2.0.1	2.0+1.6.1	2001	PSF	yes
2.1.1	2.1+2.0.1	2001	PSF	yes
2.2	2.1.1	2001	PSF	yes
2.1.2	2.1.1	2002	PSF	yes
2.1.3				
2.1.2	2002	PSF	yes	
2.2.1	2.2	2002	PSF	yes
2.2.2	2.2.1	2002	PSF	yes
2.2.3	2.2.2	2003	PSF	yes
2.3	2.2.2	2002-2003	PSF	yes
2.3.1	2.3	2002-2003	PSF	yes
2.3.2	2.3.1	2002-2003	PSF	yes
2.3.3	2.3.2	2002-2003	PSF	yes

2.3.4	2.3.3	2004	PSF	yes
2.3.5	2.3.4	2005	PSF	yes
2.4	2.3	2004	PSF	yes
2.4.1	2.4	2005	PSF	yes
2.4.2	2.4.1	2005	PSF	yes
2.4.3	2.4.2	2006	PSF	yes
2.4.4	2.4.3	2006	PSF	yes
2.5	2.4	2006	PSF	yes
2.5.1	2.5	2007	PSF	yes
2.5.2	2.5.1	2008	PSF	yes
2.5.3	2.5.2	2008	PSF	yes
2.6	2.5	2008	PSF	yes
2.6.1	2.6	2008	PSF	yes
2.6.2	2.6.1	2009	PSF	yes
2.6.3	2.6.2	2009	PSF	yes
2.6.4	2.6.3	2009	PSF	yes
2.6.5	2.6.4	2010	PSF	yes
3.0	2.6	2008	PSF	yes
3.0.1	3.0	2009	PSF	yes
3.1	3.0.1	2009	PSF	yes
3.1.1	3.1	2009	PSF	yes
3.1.2	3.1	2010	PSF	yes
3.2	3.1	2010	PSF	yes

Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

=====

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.
2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010 Python Software Foundation; All Rights Reserved" are retained in Python alone or in any derivative version prepared by Licensee.
3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.
4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.
5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.
6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.
7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.
8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3.
BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at <http://www.pythonlabs.com/logos.html> may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the Internet using the following URL: <http://hdl.handle.net/1895.22/1013>".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS

A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1,
OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6.

This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam,
The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

UNRECOGNIZED LICENSE; MD5 sum: 8ad4ae5e46a2a3afa27316325386f1cb

UNRECOGNIZED LICENSE; MD5 sum: 45ea32bdebfc2e63ec2b131cd4bc11af

UNRECOGNIZED LICENSE; MD5 sum: c197738b997df284a755c01a887f0ed9

UNRECOGNIZED LICENSE; MD5 sum: 804cd29e64bdef8111fd6b88180e19dc

UNRECOGNIZED LICENSE; MD5 sum: b5e84cb1dddb61a4f3868fc707971432

UNRECOGNIZED LICENSE; MD5 sum: 80d13bfe019695adda309f6a182a2695

UNRECOGNIZED LICENSE; MD5 sum: efc9c2dbdd704e7546d437a42525644c

License

A copyright notice accompanies this license document that identifies the copyright holders.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions in source code must retain the accompanying copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the accompanying copyright notice, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Names of the copyright holders must not be used to endorse or promote products derived from this software without prior written permission from the copyright holders.
4. If any files are modified, you must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

Disclaimer

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED

TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

UNRECOGNIZED LICENSE; MD5 sum: c4d2ef9007518f78fd9ae6bb89c51ef7

UNRECOGNIZED LICENSE; MD5 sum: ef8d703a4d226a68185ce4c5c15d90d7

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN

THE SOFTWARE.

MIT License

Copyright (c) 2010-202x The platformdirs developers

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT

OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

UNRECOGNIZED LICENSE; MD5 sum: 95594488233717b3db751ed31d203303

Copyright (c) Donald Stufft and individual contributors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

PK'u5ZtGH#license.dist-0.1.dist-info/METADATAUT ggux]AN0E>rT*VF]D
(&CjxR5[(Xf\$~"N.vVw5u%Q?8SC'%%q]#n?H]]@6=}))V_Li
^8r&ZOsyh8P]/Gy\8]C)'
c?R:@W5{Sha.PK*u5ZP%r'!license.dist-0.1.dist-info/RECORDUT ggux}N0I`0W ROonkE0UFqjgd[><i\$

)r^7f!{-"3T
]l5w|lQ9!/?2/7apt81hg\N8|15![A~&x%>PK5d#Ahr6s%license.dist-0.1.dist-info/RECORD.jwsUT
DPmgux%N@EHq()04@:*`w].9?.\$mZK'3'
}g7.hl[zMeH4 TQzI9 +g\v7+L? iOU{RC3;>>p_,7:N-f@]i#
;_*bl-X,sZzoI]4a%JQy+KI(:d=^Su`t5U@d"R7PK
d#A~XY@@ license.dist-0.1.dist-info/WHEELUT FDPmguxWheel-Version: 0.1
Packager: bdist_wheel
Root-Is-Purelib: true

PK
o@licensedist/__init__.pyUT PBmguxPK'u5ZtGH#license.dist-0.1.dist-
info/METADATAUTguxPK*u5ZP%r'!<license.dist-0.1.dist-info/RECORDUTguxPK5d#Ahr6s%ilicense.dist-
0.1.dist-info/RECORD.jwsUTDPuxPK
d#A~XY@@ license.dist-0.1.dist-info/WHEELUTFDPUxPK
o@licensedist/__init__.pyUTPUxPK
UNRECOGNIZED LICENSE; MD5 sum: 19a553a1ca654d9e1b3a5c06882c91fa
UNRECOGNIZED LICENSE; MD5 sum: c1872b37e59a0ddda1fe4cf8a81192a

UNRECOGNIZED LICENSE; MD5 sum: c628ada423b19fd269a474e38eb93287
Copyright 2012-2021 Eric Larson

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.19 libpsl 0.21.5-r3

1.19.1 Available under license :

Copyright (C) 2014-2024 Tim Rhsen

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the "Software"),
to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.

* The following License is for the source code files
psl-make-dafsa and lookup_string_in_fixed_set.c.

```
// Copyright 2015 The Chromium Authors. All rights reserved.  
//  
// Redistribution and use in source and binary forms, with or without  
// modification, are permitted provided that the following conditions are  
// met:  
//
```

```
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
// BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

1.20 mpdecimal 4.0.1-r0

1.20.1 Available under license :

Copyright (c) 2008-2025 Stefan Krah. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT

LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

SOURCE CODE LICENSE

=====

Copyright (c) 2008-2025 Stefan Krah. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

DOCUMENTATION LICENSE

=====

Copyright 2010-2025 Stefan Krah. All rights reserved.

Redistribution and use in source and 'compiled' forms (HTML, PDF, PostScript and so forth) with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer as the first lines of this file unmodified.
2. Modified documents must carry a notice that modification has occurred. This notice must also be present in any compiled form.

3. Redistributions in compiled form (converted to HTML, PDF, PostScript and other formats) must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS DOCUMENTATION IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.21 jaraco-functools 4.0.1

1.21.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.22 heimdal 18.0.2

1.22.1 Available under license :

Copyright (c) 1995 - 2014 Kungliga Tekniska Hgskolan
(Royal Institute of Technology, Stockholm, Sweden).
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Please see info documentation for the complete list of licenses.

Copyright (c) 1992, 1993

The Regents of the University of California. All rights reserved.

This code is derived from software contributed to Berkeley by
Christos Zoulas of Cornell University.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
@macro copynext{  
@vskip 20pt plus 1fil  
@end macro
```

```
@macro copyrightstart{  
@end macro
```

```
@macro copyrightend{  
@end macro
```

```
@node Copyrights and Licenses, , Acknowledgments, Top  
@comment node-name, next, previous, up  
@appendix Copyrights and Licenses
```

```
@heading Kungliga Tekniska Hgskolan
```

```
@copyrightstart  
@verbatim
```

Copyright (c) 1997-2011 Kungliga Tekniska Hgskolan
(Royal Institute of Technology, Stockholm, Sweden).
All rights reserved.

Portions Copyright (c) 2009 Apple Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the

documentation and/or other materials provided with the distribution.

3.

Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

@end verbatim

@copynext

@heading Massachusetts Institute of Technology

The

parts of the libtelnet that handle Kerberos.

@verbatim

Copyright (C) 1990 by the Massachusetts Institute of Technology

Export of this software from the United States of America may require a specific license from the United States Government.

It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of M.I.T. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. M.I.T. makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

@end

verbatim
@copynext

@heading The Regents of the University of California

The parts of the libroken, most of libtelnet, telnet, ftp,
and popper.

@verbatim

Copyright (c) 1988, 1990, 1993

The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors
may be used to endorse or promote products derived from this software
without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS
``AS IS" AND

ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

@end verbatim
@copynext

@heading The Regents of the University of California.

libedit

@verbatim

Copyright (c) 1992, 1993

The Regents of the University of California. All rights reserved.

This code is derived from software contributed to Berkeley by
Christos Zoulas of Cornell University.

Redistribution

and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

@end verbatim

@copynext

@heading TomsFastMath / LibTomMath

Tom's fast math (bignum support) and LibTomMath

@verbatim

LibTomMath is hereby released into the Public Domain.

@end verbatim

@copynext

@heading Doug Rabson

GSS-API mechlue layer.

@verbatim

Copyright (c) 2005 Doug Rabson

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

@end verbatim

@copynext

@heading PADL Software Pty Ltd

@table

@asis

@item GSS-API CFX, SPNEGO, naming extensions, API extensions.

@item KCM credential cache.

@item HDB LDAP backend.

@end table

@verbatim

Copyright (c) 2003-2011, PADL Software Pty Ltd.

Copyright (c) 2004, Andrew Bartlett.

Copyright (c) 2003 - 2008, Kungliga Tekniska Hgskolan

Copyright (c) 2015, Timothy Pearson.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of PADL Software nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY PADL SOFTWARE AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL PADL SOFTWARE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

@end verbatim

@copynext

@heading Marko Kreen

Fortuna in libhcrypto

@verbatim

Copyright (c) 2005 Marko Kreen

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

@end verbatim

@copynext

@heading NTT (Nippon Telegraph and Telephone Corporation)

Camellia in libcrypto

@verbatim

Copyright (c) 2006,2007

NTT (Nippon Telegraph and Telephone Corporation) . All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer as the first lines of this file unmodified.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY NTT ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL NTT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

@end verbatim

@copynext

@heading The NetBSD Foundation, Inc.

vis.c in libroken

@verbatim

Copyright (c) 1999, 2005 The NetBSD Foundation, Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

@end

verbatim

@copynext

@heading Vincent Rijmen, Antoon Bosselaers, Paulo Barreto

AES in libhcrypto

@verbatim

rijndael-alg-fst.c

@version 3.0 (December 2000)

Optimised ANSI C code for the Rijndael cipher (now AES)

@author Vincent Rijmen <vincent.rijmen@esat.kuleuven.ac.be>

@author Antoon Bosselaers <antoon.bosselaers@esat.kuleuven.ac.be>

@author Paulo Barreto <paulo.barreto@terra.com.br>

This code is hereby placed in the public domain.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

@end verbatim

@copynext

@heading Apple, Inc

kdc/announce.c

@verbatim

Copyright (c) 2008 Apple Inc. All Rights Reserved.

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Apple Inc. not be used in advertising or publicity pertaining

to distribution of the software without specific, written
prior
permission. Apple Inc. makes no representations about the suitability of
this software for any purpose. It is provided "as is" without express
or implied warranty.

THIS SOFTWARE IS PROVIDED ``AS IS" AND WITHOUT ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

@end verbatim

@copynext

@heading Richard Outerbridge

DES core in libcrypto

@verbatim

D3DES (V5.09) -

A portable, public domain, version of the Data Encryption Standard.

Written with Symantec's THINK (Lightspeed) C by Richard Outerbridge.
Thanks to: Dan Hoey for his excellent Initial and Inverse permutation
code; Jim Gillogly & Phil Karn for the DES key schedule code; Dennis
Ferguson, Eric Young and Dana How for comparing notes; and Ray Lau,
for humouring me on.

Copyright (c) 1988,1989,1990,1991,1992 by Richard Outerbridge.
(GENIE : OUTER; CIS : [71755,204]) Graven Imagery, 1992.

@end verbatim

@copynext

@heading
Secure Endpoints Inc

Windows support

@verbatim

Copyright (c) 2009-2015, Secure Endpoints Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

@end verbatim

@copynext

@heading Novell, Inc

lib/hcrypto/test_dh.c

@verbatim

Copyright (c) 2007, Novell, Inc.

Author: Matthias Koenig <mkoenig@suse.de>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice,

this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the Novell nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

@end verbatim

@copyrightend

1.23 apr-util 1.6.3-r1

1.23.1 Available under license :

Apache Portable Runtime Utility Library
Copyright (c) 2000-2023 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Portions of this software were developed at the National Center
for Supercomputing Applications (NCSA) at the University of
Illinois at Urbana-Champaign.

This software contains code derived from the RSA Data Security
Inc. MD5 Message-Digest Algorithm, including various
modifications by Spyglass Inc., Carnegie Mellon University, and
Bell Communications Research, Inc (Bellcore).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner

or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices

stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

APACHE PORTABLE RUNTIME SUBCOMPONENTS:

The Apache Portable Runtime includes a number of subcomponents with
separate copyright notices and license terms. Your use of the source
code for these subcomponents is subject to the terms and
conditions of the following licenses.

For the include\apr_md5.h component:

/*

* This work is derived from material Copyright RSA Data Security, Inc.

*

* The RSA copyright statement and Licence for that original material is

* included below. This is followed by the Apache copyright statement and

* licence for the modifications made to that material.

*/

/* Copyright (C) 1991-2, RSA Data Security,

Inc. Created 1991. All

rights reserved.

License to copy and use this software is granted provided that it
is identified as the "RSA Data Security, Inc. MD5 Message-Digest
Algorithm" in all material mentioning or referencing this software
or this function.

License is also granted to make and use derivative works provided
that such works are identified as "derived from the RSA Data
Security, Inc. MD5 Message-Digest Algorithm" in all material
mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either
the merchantability of this software or the suitability of this
software for any particular purpose. It is provided "as is"

without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

*/

For the passwd\apr_md5.c component:

/*

* This work is derived from material Copyright RSA Data Security, Inc.

*

* The RSA copyright statement and Licence for that original material is

* included below. This is followed by the Apache copyright statement and

* licence for the modifications made to that material.

*/

/* MD5C.C - RSA Data Security, Inc., MD5 message-digest algorithm

*/

/* Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All rights reserved.

License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

*/

/*

* The apr_md5_encode() routine uses much code obtained from the FreeBSD 3.0

* MD5 crypt() function, which is licenced as follows:

* -----

* "THE BEER-WARE LICENSE" (Revision 42):

* <phk@login.dknet.dk> wrote this file. As long as you retain this notice you

* can do whatever you want with this stuff. If we meet some day, and you think

* this stuff is worth it, you can buy me a beer in return. Poul-Henning Kamp

* -----

*/

For the crypto\apr_md4.c component:

* This is derived from material copyright RSA Data Security, Inc.

* Their notice is reproduced below in its entirety.

*

* Copyright (C) 1991-2, RSA Data Security, Inc. Created
1991. All

* rights reserved.

*

* License to copy and use this software is granted provided that it
* is identified as the "RSA Data Security, Inc. MD4 Message-Digest
* Algorithm" in all material mentioning or referencing this software
* or this function.

*

* License is also granted to make and use derivative works provided
* that such works are identified as "derived from the RSA Data
* Security, Inc. MD4 Message-Digest Algorithm" in all material
* mentioning or referencing the derived work.

*

* RSA Data Security, Inc. makes no representations concerning either
* the merchantability of this software or the suitability of this
* software for any particular purpose. It is provided "as is"
* without express or implied warranty of any kind.

*

* These notices must be retained in any copies of any part of this
* documentation and/or software.

*/

For the crypto\crypt_blowfish.c(.h) component:

* Written by Solar Designer <solar at openwall.com> in 1998-2011.

* No copyright

is claimed, and the software is hereby placed in the public

* domain. In case this attempt to disclaim copyright and place the software

* in the public domain is deemed null and void, then the software is

* Copyright (c) 1998-2011 Solar Designer and it is hereby released to the

* general public under the following terms:

*

* Redistribution and use in source and binary forms, with or without
* modification, are permitted.

*

* There's ABSOLUTELY NO WARRANTY, express or implied.

See crypto/crypt_blowfish.c for more information.

For the include\apr_md4.h component:

*
* This is derived from material copyright RSA Data Security, Inc.
* Their notice is reproduced below in its entirety.
*
* Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All
* rights reserved.
*
* License to copy and use this software is granted provided that it
* is identified as the "RSA Data Security, Inc. MD4 Message-Digest
* Algorithm" in all material mentioning or referencing this
software
* or this function.
*
* License is also granted to make and use derivative works provided
* that such works are identified as "derived from the RSA Data
* Security, Inc. MD4 Message-Digest Algorithm" in all material
* mentioning or referencing the derived work.
*
* RSA Data Security, Inc. makes no representations concerning either
* the merchantability of this software or the suitability of this
* software for any particular purpose. It is provided "as is"
* without express or implied warranty of any kind.
*
* These notices must be retained in any copies of any part of this
* documentation and/or software.
*/

For the test\testmd4.c component:

*
* This is derived from material copyright RSA Data Security, Inc.
* Their notice is reproduced below in its entirety.
*
* Copyright (C) 1990-2, RSA Data Security, Inc. Created 1990. All
* rights reserved.
*
* RSA Data Security, Inc. makes no representations concerning either
* the merchantability of this software
or the suitability of this
* software for any particular purpose. It is provided "as is"
* without express or implied warranty of any kind.
*
* These notices must be retained in any copies of any part of this
* documentation and/or software.
*/

For the xml\expat\conftools\install-sh component:

```
#
# install - install a program, script, or datafile
# This comes from X11R5 (mit/util/scripts/install.sh).
#
# Copyright 1991 by the Massachusetts Institute of Technology
#
# Permission to use, copy, modify, distribute, and sell this software and its
# documentation for any purpose is hereby granted without fee, provided that
# the above copyright notice appear in all copies and that both that
# copyright notice and this permission notice appear in supporting
# documentation, and that the name of M.I.T. not be used in advertising or
# publicity pertaining to distribution of the software without specific,
# written prior permission. M.I.T. makes no representations about the
# suitability
# of this software for any purpose. It is provided "as is"
# without express or implied warranty.
#
```

For the expat xml parser component:

Copyright (c) 1998, 1999, 2000 Thai Open Source Software Center Ltd
and Clark Cooper

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
AND NONINFRINGEMENT.

IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

For the ldap/apr_ldap_url.c component:

```
/* Portions Copyright 1998-2002 The OpenLDAP Foundation
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted only as authorized by the OpenLDAP
 * Public License. A copy of this license is available at
 * http://www.OpenLDAP.org/license.html or in file LICENSE in the
 * top-level directory of the distribution.
 *
 * OpenLDAP is a registered trademark of the OpenLDAP Foundation.
 *
 * Individual files and/or contributed packages may be copyright by
 * other parties and subject to additional restrictions.
 *
 * This work is
 * derived from the University of Michigan LDAP v3.3
 * distribution. Information concerning this software is available
 * at: http://www.umich.edu/~dirsvcs/ldap/
 *
 * This work also contains materials derived from public sources.
 *
 * Additional information about OpenLDAP can be obtained at:
 * http://www.openldap.org/
 */

/*
 * Portions Copyright (c) 1992-1996 Regents of the University of Michigan.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms are permitted
 * provided that this notice is preserved and that due credit is given
 * to the University of Michigan at Ann Arbor. The name of the University
 * may not be used to endorse or promote products derived from this
 * software without specific prior written permission. This software
 * is provided ``as is" without express or implied warranty.
 */
```

1.24 zlib 1.3.1-r2

1.24.1 Available under license :

Copyright notice:

(C) 1995-2022 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied

warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly Mark Adler
jloup@gzip.org madler@alumni.caltech.edu
Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS

FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.25 gmp 6.3.0-r3

1.25.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we

want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1

above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable

source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the

Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in

certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY

YOU OR

THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU

General Public License as published by

the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute

it

under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate

parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this

is what you want to do, use the GNU Lesser General
Public License instead of this License.

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users. We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors. You can apply it to
your programs, too.

When we speak of free software, we are referring to
freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1)

assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand

ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of

interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The

"System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not

convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to

produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years

and valid for as

long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular

product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the

entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains

a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work

occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant

patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this

License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory

patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or
that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the

GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS"

WITHOUT WARRANTY

OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17.

Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>  
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1)

assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand

ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and

"recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work

in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The

"System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system

(if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed;

section 10
makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately
publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section

7. This requirement modifies the requirement in section 4 to

"keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from

a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work,

for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on

those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions;

the above
requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and

propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a

party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this

License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS"

WITHOUT WARRANTY

OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17.

Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where

the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>

This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<https://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <<https://www.gnu.org/philosophy/why-not-lgpl.html>>.

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application

that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

- a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or
- b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

- a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

- a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the Combined Work with a copy of the GNU GPL and this license document.
- c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the

copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

1.26 libcap-ng 0.8.5-r0

1.26.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

^L

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that

any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

^L

GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it,

in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of

this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

^L

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the

complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a

work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.
^L

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever

changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2)

will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

^L

7. You may place library facilities that are a work based on the

Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined

with any other library

facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited

by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

^L

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not

excuse you from the conditions of this License. If you cannot

distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by

the Free Software Foundation.

^L

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

^L

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source

file to most
effectively convey the exclusion of warranty; and each file should
have at least the "copyright" line and a pointer to where the full
notice is found.

<one line to give the library's name and a brief idea of what it
does.>

Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor,
Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper
mail.

You should also get your employer (if you work as a programmer) or
your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James
Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any

patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole

or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete

machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each

time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates

the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.

EXCEPT WHEN

OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN

IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License,
or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

1.27 libnetfilter-queue 1.0.5-r2

1.27.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

675 Mass Ave, Cambridge, MA 02139, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of

running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you

distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6.

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and

of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) 19yy <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c'
should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.

1.28 speexdsp 1.2.1-r2

1.28.1 Available under license :

Copyright 2002-2008 Xiph.org Foundation
Copyright 2002-2008 Jean-Marc Valin
Copyright 2005-2007 Analog Devices Inc.

Copyright 2005-2008 Commonwealth Scientific and Industrial Research
Organisation (CSIRO)
Copyright 1993, 2002, 2006 David Rowe
Copyright 2003 EpicGames
Copyright 1992-1994 Jutta Degener, Carsten Bormann

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

- Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
- Neither the name of the Xiph.org Foundation nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.29 sqlite 3.49.2-r1

1.29.1 Available under license :

The author disclaims copyright to this source code. In place of
a legal notice, here is a blessing:

May you do good and not evil.
May you find forgiveness for yourself and forgive others.
May you share freely, never taking more than you give.

License Information

=====

SQLite Is Public Domain

The SQLite source code, including all of the files in the directories listed in the bullets below are

[Public Domain](https://sqlite.org/copyright.html).

The authors have submitted written affidavits releasing their work to the public for any use. Every byte of the public-domain code can be traced back to the original authors. The files of this repository that are public domain include the following:

- * All of the primary SQLite source code files found in the [src/ directory](https://sqlite.org/src/tree/src?type=tree&expand)
- * All of the test cases and testing code in the [test/ directory](https://sqlite.org/src/tree/test?type=tree&expand)
- * All of the SQLite extension source code and test cases in the [ext/ directory](https://sqlite.org/src/tree/ext?type=tree&expand)
- * All code that ends up in the "sqlite3.c" and "sqlite3.h" build products that actually implement the SQLite RDBMS.
- * All of the code used to compile the [command-line interface](https://sqlite.org/cli.html)
- * All of the code used to build various utility programs such as "sqldiff", "sqlite3_rsync", and "sqlite3_analyzer".

The public domain source files usually contain a header comment similar to the following to make it clear that the software is public domain.

> ~~~

The author disclaims copyright to this source code. In place of a legal notice, here is a blessing:

- * May you do good and not evil.
- * May you find forgiveness for yourself and forgive others.
- * May you share freely, never taking more than you give.

~~~~

Almost every file you find in this source repository will be public domain. But there are a small number of exceptions:

Non-Public-Domain Code Included With This Source Repository AS A Convenience

-----

This repository contains a (relatively) small amount of non-public-domain code

used to help implement the configuration and build logic. In other words, there are some non-public-domain files used to implement:

```
> ~~~
./configure && make
~~~
```

In all cases, the non-public-domain files included with this repository have generous BSD-style licenses. So anyone is free to use any of the code in this source repository for any purpose, though attribution may be required to reuse or republish the configure and build scripts. None of the non-public-domain code ever actually reaches the build products, such as "sqlite3.c", however, so no attribution is required to use SQLite itself. The non-public-domain code consists of scripts used to help compile SQLite. The non-public-domain code is technically not part of SQLite. The non-public-domain code is included in this repository as a convenience to developers, so that those who want to build SQLite do not need to go download a bunch of third-party build scripts in order to compile SQLite.

Non-public-domain  
code included in this repository includes:

- \* The ["autosetup"](<http://msteveb.github.io/autosetup/>) configuration system that is contained (mostly) the autosetup/ directory, but also includes the "./configure" script at the top-level of this archive. Autosetup has a separate BSD-style license. See the [autosetup/LICENSE](<http://msteveb.github.io/autosetup/license/>) for details.
- \* There are BSD-style licenses on some of the configuration software found in the legacy autoconf/ directory and its subdirectories.

The following unix shell command is can be run from the top-level of this source repository in order to remove all non-public-domain code:

```
> ~~~
rm -rf configure autosetup autoconf
~~~
```

If you unpack this source repository and then run the command above, what is left will be 100% public domain.  
Unless explicitly stated, all files which form part of autosetup are released under the following license:

-----  
autosetup - A build environment "autoconfigurator"

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE WORKWARE SYSTEMS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL WORKWARE SYSTEMS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of WorkWare Systems.

## 1.30 libmnl 1.0.5-r2

### 1.30.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE  
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software

patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the

users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

## GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's



complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based

on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary  
GNU General Public

License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or

linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany

the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

- b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not

excuse

you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a

license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

##### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is

safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!

## 1.31 nftables 1.1.3-r0

### 1.31.1 Available under license :

Original author of nftables distributed the code under the terms of the GPL version 2 \*only\*. New code though is moving to GPL version 2 or any later which is the preferred license for this project these days.

Pablo Neira Ayuso <pablo@netfilter.org>

-----



## GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's

software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we

want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and

distribute

such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License.  
(Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative

works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9.

The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE,

BE LIABLE TO YOU FOR DAMAGES,  
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING  
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED  
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY  
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER  
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE  
POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this  
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type ``show w'`.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type ``show c'` for details.

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This  
General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

## 1.32 libgcrypt 1.10.3-r1

### 1.32.1 Available under license :

GNU GENERAL PUBLIC LICENSE  
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for  
software and other kinds of works.

The licenses for most software and other practical works are designed  
to take away your freedom to share and change the works. By contrast,  
the GNU General Public License is intended to guarantee your freedom to  
share and change all versions of a program--to make sure it remains free  
software for all its users. We, the Free Software Foundation, use the



GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of

software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS

### 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To

"modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices"

to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If

the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

## 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

## 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content,

constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

### 3. Protecting Users' Legal

Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

### 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep

intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

#### 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

#### 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer

support for that product

model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge.

You need not require recipients to copy the

Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute

modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a

requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

## 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that



material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

## 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your

receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

#### 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

#### 10. Automatic Licensing of Downstream Recipients.

Each time

you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not

impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

#### 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended

to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

#### 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

#### 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

#### 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

#### 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

#### 16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF

SUCH DAMAGES.

#### 17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>

This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<http://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read  
<<http://www.gnu.org/philosophy/why-not-lgpl.html>>.

#### GNU LESSER GENERAL PUBLIC LICENSE Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>  
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

#### 0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

#### 1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

#### 2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

- a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or
- b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

#### 3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

- a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are



covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

#### 4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the

Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

## 5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
- b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

## 6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

## GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so

that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections

1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and

all its terms and conditions for copying, distributing  
or modifying  
the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the

original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR



THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute  
it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may

be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this

is what you want to do, use the GNU Lesser General  
Public License instead of this License.

KSBA is distributed under mixed GPL and LGPL licenses. Please see the file AUTHOR for details. The text of the used licenses can be found in the files:

COPYING.LGPLv3  
COPYING.GPLv3  
COPYING.GPLv2

## 1.33 git 2.49.1-r0

### 1.33.1 Available under license :

BSD License

Copyright (c) 2020, Google LLC  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google LLC nor the names of its contributors may

be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Note that the only valid version of the GPL as far as this project is concerned is `_this_` particular version of the license (ie v2, not v2.2 or v3.x or whatever), unless explicitly otherwise stated.

HOWEVER, in order to allow a migration to GPLv3 if that seems like a good idea, I also ask that people involved with the project make their preferences known. In particular, if you trust me to make that decision, you might note so in your copyright message, ie something like

This file is licensed under the GPL v2, or a later version at the discretion of Linus.

might avoid issues. But we can also just decide to synchronize and contact all copyright holders on record if/when the occasion arises.

Linus Torvalds

-----  
GNU GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute  
verbatim copies  
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your

freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means

either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices

that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands

interactively

when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable

source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means

all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions.

You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If

any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to



address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it

to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program

`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

ISC License

Copyright (c) 2011-2015 Vicent Marti

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed

to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it,

either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies  
of the Program's

source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole

or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to

control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent

license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.



10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.

#### EXCEPT WHEN

OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN

IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS

FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Note: At present, there is no interoperability between SHA-256 repositories and SHA-1 repositories.

Historically, we warned that SHA-256 repositories may later need backward incompatible changes when we introduce such interoperability features. Today, we only expect compatible changes. Furthermore, if such changes prove to be necessary, it can be expected that SHA-256 repositories created with today's Git will be usable by future versions of Git without data loss.

MIT License

Copyright (c) 2017:

Marc Stevens  
Cryptology Group  
Centrum Wiskunde & Informatica  
P.O. Box 94079, 1090 GB Amsterdam, Netherlands  
marc@marc-stevens.nl

Dan Shumow  
Microsoft Research  
danshu@microsoft.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.34 net-snmp 5.9.4-r1

### 1.34.1 Available under license :

Various copyrights apply to this package, listed in various separate parts below. Please make sure that you read all the parts.

----- Part 1: Sparta, Inc (BSD) -----

Copyright (c) 2003-2010, Sparta, Inc  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Sparta, Inc nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS

OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,

PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---- Part 2: ScienceLogic, LLC (BSD) ----

Copyright (c) 2006, ScienceLogic, LLC  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of ScienceLogic, LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Various copyrights apply to this package, listed in various separate parts below. Please make sure that you read all the parts.

---- Part 1: CMU/UCD copyright notice: (BSD like) ----

Copyright 1989, 1991, 1992 by Carnegie Mellon University

Derivative Work - 1996, 1998-2000  
Copyright 1996, 1998-2000 The Regents of the University of California

All Rights Reserved

Permission to use, copy, modify and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of CMU and The Regents of the University of California not be used in advertising or publicity pertaining to distribution of the software without specific written permission.

CMU AND THE REGENTS OF THE UNIVERSITY OF CALIFORNIA DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES

OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL CMU OR THE REGENTS OF THE UNIVERSITY OF CALIFORNIA BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM THE LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

---- Part 2: Networks Associates Technology, Inc copyright notice (BSD) -----

Copyright (c) 2001-2003, Networks Associates Technology, Inc  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of the Networks Associates Technology, Inc nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,

THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---- Part 3: Cambridge Broadband Ltd. copyright notice (BSD) ----

Portions of this code are copyright (c) 2001-2003, Cambridge Broadband Ltd.  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* The name of Cambridge Broadband Ltd. may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---- Part 4: Sun Microsystems, Inc. copyright notice (BSD) ----

Copyright (c) 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Use is subject to license terms below.

This distribution may include materials developed by third parties.

Sun, Sun Microsystems, the Sun logo and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- \* Neither the name of the Sun Microsystems, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---- Part 5: Sparta, Inc copyright notice (BSD) ----

Copyright (c) 2003-2013, Sparta, Inc  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.



\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\* Neither the name of Sparta, Inc nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

----- Part 6: Cisco/BUPTNIC copyright notice (BSD) -----

Copyright (c) 2004, Cisco, Inc and Information Network  
Center of Beijing University of Posts and Telecommunications.  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

\* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\* Neither the name of Cisco, Inc, Beijing University of Posts and Telecommunications, nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS  
PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR

CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---- Part 7: Fabasoft R&D Software GmbH & Co KG copyright notice (BSD) ----

Copyright (c) Fabasoft R&D Software GmbH & Co KG, 2003

oss@fabasoft.com

Author: Bernhard Penz <bernhard.penz@fabasoft.com>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* The name of Fabasoft R&D Software GmbH & Co KG or any of its subsidiaries, brand or product names may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---- Part 8: Apple Inc. copyright notice (BSD) ----

Copyright (c) 2007 Apple Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions

are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of Apple Inc. ("Apple") nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY APPLE AND ITS CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL APPLE OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---- Part 9: ScienceLogic, LLC copyright notice (BSD) ----

Copyright  
(c) 2009, ScienceLogic, LLC  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of ScienceLogic, LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS

“AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---- Part 10: Lennart Poettering copyright notice (BSD-like) ----

Copyright 2010 Lennart Poettering

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

---- Part 11: IETF copyright notice (BSD) ----

Copyright (c) 2013 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code

must retain the above copyright notice,  
this list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\* Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---- Part 12: Arista Networks copyright notice (BSD) ----

Copyright (c) 2013, Arista Networks, Inc.  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

\* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\* Neither the name of Arista Networks, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT

LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---- Part 13: VMware, Inc. copyright notice (BSD) ----

Copyright (c) 2016, VMware, Inc.  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of VMware, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---- Part 14: USC/Information Sciences Institute copyright notice (BSD) ----

Copyright (c) 2017-2018, Information Sciences Institute  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Information Sciences Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#### LICENSE ISSUES

=====

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact [openssl-core@openssl.org](mailto:openssl-core@openssl.org).

#### OpenSSL License

-----

/\* =====

\* Copyright (c) 1998-2008 The OpenSSL Project. All rights reserved.

\*

\* Redistribution and use in source and binary forms, with or without

\* modification, are permitted provided that the following conditions

\* are met:

\*

- \* 1. Redistributions of source code must retain the above copyright
- \* notice, this list of conditions and the following disclaimer.
- \*
- \* 2. Redistributions in binary form must reproduce the above copyright
- \* notice, this list of conditions and the following disclaimer
- in
- \* the documentation and/or other materials provided with the
- \* distribution.
- \*
- \* 3. All advertising materials mentioning features or use of this
- \* software must display the following acknowledgment:
- \* "This product includes software developed by the OpenSSL Project
- \* for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"
- \*
- \* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
- \* endorse or promote products derived from this software without
- \* prior written permission. For written permission, please contact
- \* [openssl-core@openssl.org](mailto:openssl-core@openssl.org).
- \*
- \* 5. Products derived from this software may not be called "OpenSSL"
- \* nor may "OpenSSL" appear in their names without prior written
- \* permission of the OpenSSL Project.
- \*
- \* 6. Redistributions of any form whatsoever must retain the following
- \* acknowledgment:
- \* "This product includes software developed by the OpenSSL Project
- \* for use in the OpenSSL Toolkit (<http://www.openssl.org/>)"
- \*
- \* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS" AND ANY
- \* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
- \* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
- \* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
- \* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
- \* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
- \* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
- \* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
- \* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
- \* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
- \* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
- \* OF THE POSSIBILITY OF SUCH DAMAGE.
- \* =====
- \*
- \* This product includes cryptographic software written by Eric Young
- \* ([ey@cryptsoft.com](mailto:ey@cryptsoft.com)). This product includes software
- written by Tim
- \* Hudson ([tjh@cryptsoft.com](mailto:tjh@cryptsoft.com)).
- \*
- \*/



## Original SSLeay License

```
-----

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to. The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code. The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of the parts of the
 * library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *    "This product includes cryptographic software written by
 *     Eric Young (eay@cryptsoft.com)"
 *    The word 'cryptographic' can be left out if the rouines from the library
 *    being used are not cryptographic
 *    related :-).
 * 4. If you include any Windows specific code (or a derivative thereof) from
 *    the apps directory (application code) you must include an acknowledgement:
 *    "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS" AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
```

- \* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
- \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
- \* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
- \* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
- \* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
- \* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
- \* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
- OF
- \* SUCH DAMAGE.
- \*
- \* The licence and distribution terms for any publically available version or
- \* derivative of this code cannot be changed. i.e. this code cannot simply be
- \* copied and put under another distribution licence
- \* [including the GNU Public Licence.]
- \*/

## 1.35 htop 3.4.1-r0

### 1.35.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program

is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based

on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering

access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to

apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

## NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE,  
THERE IS NO WARRANTY

FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of



MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute
it
under certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

## 1.36 alpine-base 3.22.2-r0

### 1.36.1 Available under license :

# MIT License

Copyright 2019-2021 Caleb Porzio and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.  
MIT

## 1.37 zipp 3.19.2

### 1.37.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.38 futures 3.12

## 1.38.1 Available under license :

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

- 
1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.
  2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006 Python Software Foundation; All Rights Reserved" are retained in Python alone or in any derivative version prepared by Licensee.
  3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.
  4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.
  5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.
  6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.
  7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

## 1.39 libpkit 0.25.5-r2

### 1.39.1 Available under license :

pygit2 is Copyright (C) the pygit2 contributors,  
unless otherwise stated. See the AUTHORS.md file for details.

Note that the only valid version of the GPL as far as this project is concerned is `_this_` particular version of the license (ie v2, not v2.2 or v3.x or whatever), unless explicitly otherwise stated.

---

#### LINKING EXCEPTION

In addition to the permissions in the GNU General Public License, the authors give you unlimited permission to link the compiled version of this library into combinations with other programs, and to distribute those combinations without any restriction coming from the use of this file. (The General Public License restrictions do apply in other respects; for example, they cover modification of the file, and distribution when not linked into a combined executable.)

---

#### GNU GENERAL PUBLIC LICENSE Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to

your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed

under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this

License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not



excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free

Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by

the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions;  
type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the

library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

## 1.40 popt 1.19-r4

### 1.40.1 Available under license :

Copyright (c) 1998 Red Hat Software

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.41 glib 2.84.4-r0

### 1.41.1 Available under license :

This work may be reproduced and distributed in whole or in part, in any medium, physical or electronic, so as long as this copyright notice remains intact and unchanged on all copies. Commercial redistribution is permitted and encouraged, but you may not redistribute, in whole or in part, under terms more restrictive than those under which you received it. If you redistribute a modified or translated version of this work, you must also make the source code to the modified or translated version available in electronic form without charge. However, mere aggregation as part of a larger work shall not count as a modification for this purpose.

All code examples in this work are placed into the public domain, and may be used, modified and redistributed without restriction.

BECAUSE THIS WORK IS LICENSED FREE OF CHARGE, THERE IS NO

WARRANTY FOR THE WORK, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE WORK "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. SHOULD THE WORK PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE WORK AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE WORK, EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the

Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a

NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the

NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable  
law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.  
GNU GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom  
to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.



For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for

each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under

the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of

any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work,

complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs,

unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense

or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use

of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this

License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the

Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a

pointer to where the full notice is found.

one line to give the program's name and an idea of what it does. Copyright (C) yyyy name of author

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA. Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989 Ty Coon, President of Vice  
GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully

about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you.

You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License

permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software

library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing

it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work

based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given



copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License.

(If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its

use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from

the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT

WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply

##### These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the library's name and an idea of what it does.

Copyright (C) year name of author

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in  
the library 'Frob' (a library for tweaking knobs) written  
by James Random Hacker.

signature of Ty Coon, 1 April 1990

Ty Coon, President of Vice

That's all there is to it!

This work is provided "as is"; redistribution and modification  
in whole or in part, in any medium, physical or electronic is  
permitted without restriction.

This work is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

In no event shall the authors or contributors be liable for any  
direct, indirect, incidental, special, exemplary, or consequential  
damages (including, but not limited to, procurement of substitute  
goods or services; loss of use, data, or profits; or business  
interruption) however caused and on any theory of liability, whether  
in contract, strict liability, or tort (including negligence or  
otherwise) arising in any way out of the use of this software, even  
if advised of the possibility of such damage.

MIT License

Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the

Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

Mozilla Public License Version 1.1

## 1. Definitions.

1.0.1. "Commercial Use" means distribution or otherwise making the Covered Code available to a third party.

1.1. "Contributor" means each entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. "Covered Code" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.

1.4. "Electronic Distribution Mechanism" means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. "Executable" means Covered Code in any form other than Source Code.

1.6. "Initial Developer" means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. "License" means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:  
Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.  
Any new file that contains any part of the Original Code or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

## 2. Source Code License.

2.1. The Initial Developer Grant. The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

- a. under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and
- b. under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).
- c. the licenses granted in this Section 2.1 (a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.
- d. Notwithstanding Section 2.1 (b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused by: i) the modification of the Original Code or ii) the combination of the Original Code with other software or devices.

2.2. Contributor Grant. Subject to third party intellectual property claims, each Contributor hereby grants You a

world-wide, royalty-free, non-exclusive license

a. under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and

b. under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

c. the licenses granted in Sections 2.2 (a) and 2.2 (b) are effective on the date Contributor first makes Commercial Use of the Covered Code.

d. Notwithstanding Section 2.2 (b) above, no patent license is granted: 1) for any code that Contributor has deleted from the Contributor Version; 2) separate from the Contributor Version; 3) for infringements caused by: i) third party modifications of Contributor Version or ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or 4) under Patent Claims infringed by Covered Code in the absence of Modifications made by that Contributor.

### 3. Distribution Obligations.

3.1. Application of License. The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However,

You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code. Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications. You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

### 3.4. Intellectual Property Matters

#### (a) Third Party Claims

If Contributor has knowledge that a license under a third party's intellectual property rights is required to



exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If Contributor obtains such knowledge after the Modification is made

available as described in Section 3.2, Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

(b) Contributor APIs

If Contributor's Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file.

(c) Representations.

Contributor represents that, except as disclosed pursuant to Section 3.4 (a) above, Contributor believes that Contributor's Modifications are Contributor's original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by this License.

3.5. Required Notices. You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.6. Distribution of Executable Versions. You may distribute Covered Code in Executable form only if the requirements of Sections 3.1, 3.2, 3.3, 3.4 and 3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.7. Larger Works. You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

#### 4. Inability to Comply Due to Statute or Regulation.

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

#### 5. Application of this License.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.

#### 6. Versions of the License.

##### 6.1. New Versions

Netscape Communications Corporation ("Netscape") may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

##### 6.2. Effect of New Versions

Once

Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Netscape. No one other than Netscape has the right to modify the terms applicable to Covered Code created under this License.

##### 6.3. Derivative Works

If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename Your license so that the phrases "Mozilla", "MOZILLAPL", "MOZPL", "Netscape", "MPL", "NPL" or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

#### 7. DISCLAIMER OF WARRANTY

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

## 8. Termination

8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that:

a. such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

b. any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version

directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

## 9. LIMITATION OF LIABILITY

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING,

WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

#### 10. U.S. government end users

The Covered Code is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

#### 11. Miscellaneous

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International

Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

#### 12. Responsibility for claims

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

#### 13. Multiple-licensed code

Initial Developer may designate portions of the Covered Code as "Multiple-Licensed". "Multiple-Licensed" means that the Initial Developer permits you to utilize portions of the Covered Code under Your choice of the MPL or the alternative licenses, if any, specified by the Initial Developer in the file described in Exhibit A.

Exhibit A - Mozilla Public License.

"The

contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.mozilla.org/MPL/>

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is \_\_\_\_\_.

The Initial Developer of the Original Code is \_\_\_\_\_.

Portions created by \_\_\_\_\_ are Copyright (C) \_\_\_\_\_  
\_\_\_\_\_. All Rights Reserved.

Contributor(s): \_\_\_\_\_.

Alternatively, the contents of this file may be used under the terms of the \_\_\_\_\_ license (the "[\_\_\_\_\_] License"), in which case the provisions of [\_\_\_\_\_] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [\_\_\_\_\_] License and not to allow others to use your version of this file under the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [\_\_\_\_\_] License. If you do not delete the provisions above, a recipient may use your version of this file under either the MPL or the [\_\_\_\_\_] License."

NOTE: The text of this Exhibit A may differ slightly from the text of the notices in the Source Code files of the Original Code. You should use the text of this Exhibit A rather than the text found in the Original Code Source Code for Your Modifications.

Creative Commons Legal Code

CC0 1.0 Universal

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS DOCUMENT DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED HEREUNDER, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED HEREUNDER.

Statement of Purpose

The laws of most jurisdictions throughout the world automatically confer exclusive Copyright and Related Rights (defined below) upon the creator and subsequent owner(s) (each and all, an "owner") of an original work of authorship and/or a database (each, a "Work").

Certain owners wish to permanently relinquish those rights to a Work for the purpose of contributing to a commons of creative, cultural and scientific works ("Commons") that the public can reliably and without fear of later claims of infringement build upon, modify, incorporate in other works, reuse and redistribute as freely as possible in any form whatsoever and for any purposes, including without limitation commercial purposes. These owners may contribute to the Commons to promote the ideal of a free culture and the further production of creative, cultural and scientific works, or to gain reputation or greater distribution for their Work in part through the use and efforts of others.

For these and/or other purposes and motivations, and without any expectation of additional consideration or compensation, the person associating CC0 with a Work (the "Affirmer"), to the extent that he or she

is an owner of Copyright and Related Rights in the Work, voluntarily elects to apply CC0 to the Work and publicly distribute the Work under its terms, with knowledge of his or her Copyright and Related Rights in the Work and the meaning and intended legal effect of CC0 on those rights.

1. Copyright and Related Rights. A Work made available under CC0 may be protected by copyright and related or neighboring rights ("Copyright and Related Rights"). Copyright and Related Rights include, but are not limited to, the following:

- i. the right to reproduce, adapt, distribute, perform, display, communicate, and translate a Work;
- ii. moral rights retained by the original author(s) and/or performer(s);
- iii. publicity and privacy rights pertaining to a person's image or likeness depicted in a Work;
- iv. rights protecting against unfair competition in regards to a Work, subject to the limitations in paragraph 4(a), below;
- v. rights protecting the extraction, dissemination, use and reuse of data in a Work;
- vi. database rights (such as those arising under Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, and under any national implementation

thereof, including any amended or successor version of such directive); and

- vii. other similar, equivalent or corresponding rights throughout the world based on applicable law or treaty, and any national implementations thereof.

2. Waiver. To the greatest extent permitted by, but not in contravention of, applicable law, Affirmer hereby overtly, fully, permanently, irrevocably and unconditionally waives, abandons, and surrenders all of Affirmer's Copyright and Related Rights and associated claims and causes of action, whether now known or unknown (including existing as well as future claims and causes of action), in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "Waiver"). Affirmer makes the Waiver for the benefit of each member of the public at large and to the detriment of Affirmer's heirs and successors, fully intending that such Waiver shall not be subject to revocation, rescission, cancellation, termination, or any other legal or equitable action to disrupt the quiet enjoyment of the Work by the public as contemplated by Affirmer's express Statement of Purpose.

3. Public License Fallback. Should any part of the Waiver for any reason be judged legally invalid or ineffective under applicable law, then the Waiver shall be preserved to the maximum extent permitted taking into account Affirmer's express Statement of Purpose. In addition, to the extent the Waiver is so judged Affirmer hereby grants to each affected person a royalty-free, non transferable, non sublicensable, non exclusive, irrevocable and unconditional license to exercise Affirmer's Copyright and Related Rights in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "License"). The License shall be deemed effective as of the date CC0 was applied by Affirmer to the Work. Should any part of the License for any reason be judged legally invalid or ineffective under applicable law, such partial invalidity or ineffectiveness shall not invalidate the remainder of the License, and in such case Affirmer hereby affirms that he or she will not (i) exercise any of his or her remaining Copyright and Related Rights in the Work or (ii) assert any associated claims and causes of action with respect to the Work, in either case contrary to Affirmer's express Statement of Purpose.

#### 4. Limitations and Disclaimers.

- a. No trademark or patent rights held by Affirmer are waived, abandoned, surrendered, licensed or otherwise affected by this document.
- b. Affirmer offers the Work as-is and makes no representations or warranties of any kind concerning the Work, express, implied, statutory or otherwise, including without limitation warranties of title, merchantability, fitness for a particular purpose, non infringement, or the absence of latent or other defects, accuracy, or the present or absence of errors, whether or not discoverable, all to the greatest extent permissible under applicable law.
- c. Affirmer disclaims responsibility for clearing rights of other persons that may apply to the Work or any use thereof, including without limitation any person's Copyright and Related Rights in the Work. Further, Affirmer disclaims responsibility for obtaining any necessary consents, permissions or other rights required for any use of the Work.
- d. Affirmer understands and acknowledges that Creative Commons is not a party to this document and has no duty or obligation with respect to this CC0 or use of the Work.

# 1.42 libnetfilter-contrack 1.4.8

## 1.42.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

675 Mass Ave, Cambridge, MA 02139, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.



Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole

or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work

based on it,  
under Section 2) in object code or executable form under the terms of  
Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6.

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING

WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),

EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute it under certain conditions; type ``show c'` for details.

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program ``Gnomovision'` (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

## 1.43 apr 1.7.5-r0

### 1.43.1 Available under license :

Apache Portable Runtime  
Copyright 2024 The Apache Software Foundation.

This product includes software developed at  
The Apache Software Foundation (<http://www.apache.org/>).

Portions of this software were developed at the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign.

This software contains code derived from the RSA Data Security Inc. MD5 Message-Digest Algorithm.

This software contains code derived from UNIX V7, Copyright(C) Caldera International Inc.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally



submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
- You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

#### APACHE PORTABLE RUNTIME SUBCOMPONENTS:

The Apache Portable Runtime includes a number of subcomponents with  
separate copyright notices and license terms. Your use of the source  
code for these subcomponents is subject to the terms and conditions  
of the following licenses.

From strings/apr\_fnmatch.c, include/apr\_fnmatch.h, misc/unix/getopt.c,  
file\_io/unix/mktemp.c, strings/apr\_strings.c:

```
/*  
 * Copyright (c) 1987, 1993, 1994  
 *   The Regents of the University of California. All rights reserved.  
 *  
 * Redistribution and use in source and binary forms, with or without  
 * modification, are permitted provided that the following conditions  
 * are met:  
 * 1. Redistributions of source code must retain the above copyright  
 *   notice, this list of conditions and the following disclaimer.  
 * 2. Redistributions in binary form must reproduce the above copyright  
 *   notice, this list of conditions and the following disclaimer in the  
 *   documentation and/or other materials provided with the distribution.  
 * 3. All advertising materials mentioning features or use of this software  
 *   must display the following acknowledgement:  
 *     This product includes software developed by the University of  
 *     California, Berkeley and its contributors.  
 * 4. Neither the name of the University nor the names of its contributors  
 *   may be used to endorse or promote products derived from this software  
 *   without specific prior written permission.  
 *  
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
```

\* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
\* IMPLIED WARRANTIES OF  
\* MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
\* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE  
\* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL  
\* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS  
\* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
\* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT  
\* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY  
\* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF  
\* SUCH DAMAGE.

From network\_io/unix/inet\_ntop.c, network\_io/unix/inet\_pton.c:

```
/* Copyright (c) 1996 by Internet Software Consortium.  
*  
* Permission to use, copy, modify, and distribute this software for any  
* purpose with or without fee is hereby granted, provided that the above  
* copyright notice and this permission notice appear in all copies.  
*  
* THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE  
* CONSORTIUM DISCLAIMS  
* ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES  
* OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE  
* CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL  
* DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR  
* PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS  
* ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS  
* SOFTWARE.
```

From dso/aix/dso.c:

```
* Based on libdl (dlfcn.c/dlfcn.h) which is  
* Copyright (c) 1992,1993,1995,1996,1997,1988  
* Jens-Uwe Mager, Helios Software GmbH, Hannover, Germany.  
*  
* Not derived from licensed software.  
*  
* Permission is granted to freely use, copy, modify, and redistribute  
* this software, provided that the author is not construed to be liable  
* for any results of using the software, alterations are clearly marked  
* as such, and this notice is not modified.
```

From strings/apr\_strnatcmp.c,  
include/apr\_strings.h:

```
strnatcmp.c -- Perform 'natural order' comparisons of strings in C.  
Copyright (C) 2000 by Martin Pool <mbp@humbug.org.au>
```

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

From  
strings/apr\_snprintf.c:

```
*  
* cvt - IEEE floating point formatting routines.  
*   Derived from UNIX V7, Copyright(C) Caldera International Inc.  
*
```

Copyright(C) Caldera International Inc. 2001-2002. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code and documentation must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

All advertising materials mentioning features or use of this software must display the following acknowledgement:

This product includes software developed or owned by Caldera International, Inc.

Neither the name of Caldera International, Inc. nor the names of other contributors may be used to endorse or promote products derived from this software without specific prior written permission.

USE OF THE SOFTWARE PROVIDED FOR UNDER THIS LICENSE BY CALDERA INTERNATIONAL, INC. AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL CALDERA INTERNATIONAL, INC. BE LIABLE FOR ANY DIRECT, INDIRECT INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.44 pcre2 10.46-r0

### 1.44.1 Available under license :

PCRE2 LICENCE

Please see the file LICENCE in the PCRE2 distribution for licensing details.

End

PCRE2 License

=====

| SPDX-License-Identifier: | BSD-3-Clause WITH PCRE2-exception |  
|-----|-----|

PCRE2 is a library of functions to support regular expressions whose syntax and semantics are as close as possible to those of the Perl 5 language.

Releases 10.00 and above of PCRE2 are distributed under the terms of the "BSD" licence, as specified below, with one exemption for certain binary redistributions. The documentation for PCRE2, supplied in the "doc" directory, is distributed under the same terms as the software itself. The data in the testdata directory is not copyrighted and is in the public domain.

The basic library functions are written in C and are freestanding. Also included in the distribution is a just-in-time compiler that can be used to optimize pattern matching. This is an optional feature that can be omitted when the library is built.

COPYRIGHT

-----

### The basic library functions

Written by: Philip Hazel  
Email local part: Philip.Hazel  
Email domain: gmail.com

Retired from University of Cambridge Computing Service,  
Cambridge, England.

Copyright (c) 1997-2007 University of Cambridge  
Copyright (c) 2007-2024 Philip Hazel  
All rights reserved.

### ### PCRE2 Just-In-Time compilation support

Written by: Zoltan Herczeg  
Email local part: hzmester  
Email domain: freemail.hu

Copyright (c) 2010-2024 Zoltan Herczeg  
All rights reserved.

### ### Stack-less Just-In-Time compiler

Written by: Zoltan Herczeg  
Email local part: hzmester  
Email domain: freemail.hu

Copyright (c) 2009-2024 Zoltan Herczeg  
All rights reserved.

### ### All other contributions

Many other contributors have participated in the authorship of PCRE2. As PCRE2 has never required a Contributor Licensing Agreement, or other copyright assignment agreement, all contributions have copyright retained by each original contributor or their employer.

### THE "BSD" LICENCE

-----

Redistribution  
and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions are met:

\* Redistributions of source code must retain the above copyright notices,  
this list of conditions and the following disclaimer.



\* Redistributions in binary form must reproduce the above copyright notices, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\* Neither the name of the University of Cambridge nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR

ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#### EXEMPTION FOR BINARY LIBRARY-LIKE PACKAGES

-----  
The second condition in the BSD licence (covering binary redistributions) does not apply all the way down a chain of software. If binary package A includes PCRE2, it must respect the condition, but if package B is software that includes package A, the condition is not imposed on package B unless it uses PCRE2 independently.

End

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,

INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.45 alpine-keys 2.5-r0

### 1.45.1 Available under license :

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE

OR OTHER DEALINGS IN THE SOFTWARE.

# Maintainer: Natanael Copa <ncopa@alpinelinux.org>

pkgname=alpine-keys

pkgver=2.5

pkgrel=0

pkgdesc="Public keys for Alpine Linux packages"

url="https://alpinelinux.org"

# we install arch specific keys to /etc so we cannot do arch=noarch

arch="all"

license="MIT"

replaces="alpine-base"

## 1.46 distro 1.9.0

### 1.46.1 Available under license :

We thank anyone who has contributed to this project, and welcome anyone to contribute further.

For a list of contributors, please see <https://github.com/python-distro/distro/graphs/contributors>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You"  
(or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

## 1.47 tcptraceroute 1.5\_beta7-r4

### 1.47.1 Available under license :

This package was debianized by Michael C. Toren <mct@toren.net> on  
Sun, 1 Jul 2001 13:33:41 -0400, from source code obtained from  
<http://michael.toren.net/code/tcptraceroute/>

Copyright:

Copyright (c) 2001-2006 Michael C. Toren <mct@toren.net>

This program is free software; you can redistribute it and/or modify it  
under the terms of the GNU General Public License, version 2, as published  
by the Free Software Foundation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY  
WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A  
PARTICULAR PURPOSE. See the GNU General Public License for more details.

A copy of the GNU GPL is available as /usr/share/common-licenses/GPL-2 on Debian  
systems, or on the World Wide Web at <http://www.gnu.org/copyleft/gpl.html> You  
can also obtain it by writing to the Free Software Foundation, Inc., 59 Temple  
Place - Suite 330, Boston, MA 02111-1307, USA

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software



patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each

time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding

those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.

EXCEPT WHEN

OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN

IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may

be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

## 1.48 libevent 2.1.12-r8

### 1.48.1 Available under license :

Libevent is available for use under the following license, commonly known as the 3-clause (or "modified") BSD license:

=====

Copyright (c) 2000-2007 Niels Provos <provos@citi.umich.edu>  
Copyright (c) 2007-2012 Niels Provos and Nick Mathewson

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

Portions of Libevent are based on works by others, also made available by them under the three-clause BSD license above. The copyright notices are available in the corresponding source files; the license is as above. Here's a list:

log.c:

Copyright (c) 2000 Dug Song <dugsong@monkey.org>  
Copyright (c) 1993 The Regents of the University of California.

strlcpy.c:

Copyright (c) 1998 Todd C. Miller <Todd.Miller@courtesan.com>

win32select.c:

Copyright (c) 2003 Michael A. Davis <mike@datanerds.net>

evport.c:

Copyright (c) 2007 Sun Microsystems

ht-internal.h:

Copyright (c) 2002 Christopher Clark

minheap-internal.h:

Copyright (c) 2006 Maxim Yegorushkin <maxim.yegorushkin@gmail.com>

=====

The arc4module is available under the following, sometimes called the "OpenBSD" license:

Copyright (c) 1996, David Mazieres <dm@uun.org>  
Copyright (c) 2008, Damien Miller <djm@openbsd.org>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR

ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN



ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

=====

The Windows timer code is based on code from libutp, which is distributed under this license, sometimes called the "MIT" license.

Copyright (c) 2010 BitTorrent, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CMake - Cross Platform Makefile Generator

Copyright 2000-2013 Kitware, Inc.

Copyright 2000-2011 Insight Software Consortium

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the names of Kitware, Inc., the Insight Software Consortium, nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written

permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,

THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-----

The above copyright and license notice applies to distributions of CMake in source and binary form. Some source files contain additional notices of original copyright by their contributors; see each source for details. Third-party software packages supplied with CMake under compatible licenses provide their own copyright notices documented in corresponding subdirectories.

-----

CMake was initially developed by Kitware with the following sponsorship:

- \* National Library of Medicine at the National Institutes of Health as part of the Insight Segmentation and Registration Toolkit (ITK).
- \* US National Labs (Los Alamos, Livermore, Sandia) ASC Parallel Visualization Initiative.
- \* National Alliance for Medical Image Computing (NAMIC) is funded by the National Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 EB005149.

\* Kitware, Inc.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the copyright notice, this list of conditions and the following disclaimer in the

documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.49 wireshark 4.4.7-r0

### 1.49.1 Available under license :

Installing Wireshark on FreeBSD/OpenBSD/NetBSD/DragonFly BSD

=====

1. Berkeley Packet Filter (BPF) requirement
2. Running Wireshark as a non-root user

For general installation instructions, see the INSTALL file, along with the Developer's Guide located at <https://www.wireshark.org/docs/> and in the doc/ directory. Additional BSD specific notes and requirements follow.

1. Berkeley Packet Filter (BPF) requirement

-----

In order to capture packets (with Wireshark/TShark, tcpdump, or any other packet capture program) on a BSD system, your kernel must have the Berkeley Packet Filter mechanism enabled. The default kernel configurations in recent versions of BSD systems have this enabled already. To verify the bpf device is present, look in the /dev directory:

```
ls -l /dev/bpf*
```

You should see one or more bpf devices listed similar to this:

```
crw----- 1 root wheel
0, 90 Aug 10 21:05 /dev/bpf0
crw----- 1 root wheel 0, 91 Aug 10 21:05 /dev/bpf1
```

Packet-capturing programs will pick the first bpf device that's not in use. Recent versions of most BSDs will create bpf devices as needed, so you don't have to configure the number of devices that will be available.

## 2. Running wireshark as a non-root user

-----  
Since the bpf devices are read-only by the owner (root), you normally have to run packet capturing programs such as Wireshark as root. It is safer to run programs as a non-root user if possible. To run Wireshark as a non-root user, you must change the permissions on the bpf device(s). If you are the only user that needs to use Wireshark, the easiest way is to change the owner of each bpf device to your username. You can also add the read/write ability to the group (typically wheel) and add users that need to use Wireshark to the wheel group. Check your operating system's documentation on how to make permanent these changes as they are often reset upon reboot; if /dev is implemented with devfs, it might be possible to configure devfs to create all bpf devices owned by a particular user and/or group and with particular permissions. In FreeBSD 6.0 and later this can be done by creating an /etc/devfs.rules file with content such as

```
[localrules=10]
add path 'bpf*' {mode and permissions}
```

where "mode and permissions" can include clauses such as

```
mode {octal permissions}
```

to set the permissions on the device (e.g., "mode 0660" to set the permissions to rw-rw-r--),

```
user {user}
```

to set the user who owns the device, or

```
group {group}
```

to set the group that owns the device and adding a line such as

```
devfs_system_ruleset=localrules
```

to /etc/rc.conf. For example, an /etc/devfs.rules file with

```
[localrules=10]
add path 'bpf*' mode 0660 group wheel
```

```

will grant read and write permissions on all BPF devices to all users in
the "wheel" group.
/* packet-epl.h
* Routines for "Ethernet POWERLINK 2.0" dissection
* (Ethernet POWERLINK V2.0 Communication Profile Specification Draft Standard Version 1.2.0)
*
* A dissector for:
* Wireshark - Network traffic analyzer
* By Gerald Combs <gerald@wireshark.org>
* Copyright 1999 Gerald Combs
*
* SPDX-License-Identifier: GPL-2.0-or-later
*/

```

```

#ifndef __EPL_H_
#define __EPL_H_

```

```

#include <epan/address.h>
#include <epan/wmem_scopes.h>
#include <epan/range.h>

```

```

struct epl_datatype;

```

```

struct profile {
    uint16_t id;
    uint8_t nodeid;
    address node_addr;

```

```

    uint32_t vendor_id;
    uint32_t product_code;

```

```

    wmem_map_t *objects;
    wmem_allocator_t *scope, *parent_scope;
    wmem_map_t *parent_map;

```

```

    char *name;
    char *path;
    void *data;
    unsigned cb_id;
    wmem_array_t *TPDO; /* CN->MN */
    wmem_array_t *RPDO; /* MN->CN */

```

```

    struct profile *next;
};

```

```

enum { OD_ENTRY_SCALAR = 7, OD_ENTRY_ARRAY = 8, OD_ENTRY_RECORD = 9 };

```

```

struct od_entry {
    uint16_t idx;

```

```

/* This is called the ObjectType in the standard,

* but this is too easy to be mistaken with the
* DataType.
* ObjectType specifies whether it's a scalar or
* an aggregate
*/
uint16_t type_class;
char name[64];
/* Called DataType by the standard,
* Can be e.g. Unsigned32
*/
const struct epl_datatype *type;
uint64_t value;
};
#define OD_ENTRY_INITIALIZER { 0, 0, { 0 }, 0, 0 }

struct subobject {
    range_admin_t range;
    struct od_entry info;
};
#define SUBOBJECT_INITIALIZER { RANGE_ADMIN_T_INITIALIZER, OD_ENTRY_INITIALIZER }

typedef struct epl_wmem_iarray epl_wmem_iarray_t;

struct object {
    struct od_entry info;
    epl_wmem_iarray_t *subindices;
};

struct profile;

const struct epl_datatype *epl_type_to_hf(const char *name);

static inline gboolean
subobject_equal(const void *_a, const void *_b)
{
    const struct od_entry *a = &((const struct subobject*)_a)->info;
    const struct od_entry *b = &((const struct subobject*)_b)->info;

    return a->type_class == b->type_class
        && a->type == b->type
        && g_str_equal(a->name,
            b->name);
}

struct profile *epl_xdd_load(struct profile *profile, const char *xml_file);

```

```

void epl_eds_init(void);
struct profile *epl_eds_load(struct profile *profile, const char *eds_file);

struct object *epl_profile_object_add(struct profile *profile, uint16_t idx);
struct object *epl_profile_object_lookup_or_add(struct profile *profile, uint16_t idx);

bool epl_profile_object_mapping_add(struct profile *profile, uint16_t idx, uint8_t subindex, uint64_t mapping);
bool epl_profile_object_mappings_update(struct profile *profile);

range_admin_t * epl_wmem_iarray_find(epl_wmem_iarray_t *arr, uint32_t value);
bool epl_wmem_iarray_is_empty(epl_wmem_iarray_t *iarr);
bool epl_wmem_iarray_is_sorted(epl_wmem_iarray_t *iarr);

#define EPL_OBJECT_MAPPING_SIZE ((unsigned)sizeof (uint64_t))

#define CHECK_OVERLAP_ENDS(x1, x2, y1, y2) ((x1) < (y2) && (y1) < (x2))
#define CHECK_OVERLAP_LENGTH(x, x_len, y, y_len) \
CHECK_OVERLAP_ENDS((x), (x) + (x_len), (y), (y) + (y_len))

#endif

```

Copyright notice for the files copied from

<https://web.archive.org/web/20080613235159/http://www.opensync.org/browser/branches/3rd-party-cmake-modules/modules>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO

EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF

THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,

<<https://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we



want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and

distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
  - b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
  - c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program
- in object code or executable form with such  
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled  
to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are

prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies  
a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM,  
TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING

OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, see <<https://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show

c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Moe Ghoul>, 1 April 1989  
Moe Ghoul, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

```
/* <filename>.c
 *
 * Wireshark - Network traffic analyzer
 * By Gerald Combs <gerald@wireshark.org>
 * Copyright 1998 Gerald Combs
 *
 * SPDX-License-Identifier: GPL-2.0-or-later
 */
```

License of Lrexlib release  
-----

Copyright (C) Reuben Thomas 2000-2020  
Copyright (C) Shmuel Zeigerman 2004-2020

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY

KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,

WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Format: <https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Upstream-Name: Wireshark

Source: <https://www.wireshark.org/>

Files-Excluded: debian

Files: \*

Copyright: Gerald Combs <[gerald@wireshark.org](mailto:gerald@wireshark.org)> and contributors

Comment: Wireshark and the "fin" logo are trademarks of Gerald Combs.

License: GPL-2+

Files: debian/\*

packaging/debian/\*

Copyright: 1998-2008 Frederic Peters <[fpeters@debian.org](mailto:fpeters@debian.org)>

2005-2009 Joost Yervante Damad <[andete@debian.org](mailto:andete@debian.org)>

2009-2017 Balint Reczey <[balint@balintreczey.hu](mailto:balint@balintreczey.hu)>

2017- Balint Reczey <[balint.reczey@canonical.com](mailto:balint.reczey@canonical.com)>

License: GPL-2+

Files: wsutil/g711.c:

Copyright: Sun Microsystems, Inc.

License: public-domain

This source code is a product of Sun Microsystems, Inc. and is provided for unrestricted use. Users may copy or modify this source code without charge.

.

SUN SOURCE CODE IS PROVIDED AS IS WITH NO WARRANTIES OF ANY KIND INCLUDING THE WARRANTIES OF DESIGN, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, OR ARISING FROM A COURSE OF DEALING, USAGE OR TRADE PRACTICE.

.

Sun source code is provided with no support and without any obligation on the part of Sun Microsystems, Inc. to assist in its use, correction, modification or enhancement.

.

SUN MICROSYSTEMS, INC. SHALL HAVE NO LIABILITY WITH RESPECT TO THE INFRINGEMENT OF COPYRIGHTS, TRADE SECRETS OR ANY PATENTS BY THIS SOFTWARE OR ANY PART THEREOF.

.

In no event will Sun Microsystems, Inc. be liable for any lost revenue or profits or other special, indirect and consequential damages, even if Sun has been advised of the possibility of such damages.



.  
Sun Microsystems, Inc.  
2550 Garcia Avenue  
Mountain View, California 94043

Files: wsutil/strptime.c  
Copyright: 1991, 1992, 1996, 1998 Free Software Foundation, Inc.  
Comment: Contributed by Ulrich Drepper <drepper@cygnus.com>, 1996  
License: LGPL-2+

Files: tools/lemon/lempar.c  
Copyright: 1991-1995 by D. Richard Hipp.  
License: LGPL-2+

Files: epan/except.c  
Copyright:  
1999 Kaz Kylheku <kaz@ashi.footprints.net>  
License: FSL-Kaz  
Free Software License:

.  
All rights are reserved by the author, with the following exceptions:  
Permission is granted to freely reproduce and distribute this software,  
possibly in exchange for a fee, provided that this copyright notice appears  
intact. Permission is also granted to adapt this software to produce  
derivative works, as long as the modified versions carry this copyright  
notice and additional notices stating that the work has been modified.  
This source code may be translated into executable form and incorporated  
into proprietary software; there is no requirement for such software to  
contain a copyright notice related to this source.

Files: epan/in\_cksum.c  
Copyright: 1988, 1992, 1993 The Regents of the University of California.  
License: BSD-3-clause  
All rights reserved.

.  
Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions  
are met:  
1. Redistributions of source code must retain the above copyright  
notice, this list of conditions and the following disclaimer.  
2. Redistributions in binary form must reproduce the above copyright  
notice, this list of conditions and the following disclaimer in the  
documentation and/or other materials provided with the distribution.  
3. Neither the name of the University nor the names of its contributors  
may be used to endorse or promote products derived from this software  
without specific prior written permission.

.  
THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND

ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files: epan/dissectors/packet-enc.c  
      epan/dissectors/packet-etherip.c  
      epan/dissectors/packet-ipsec-udp.c

Copyright: 2003 Markus Friedl

License: BSD-2-clause

Files: epan/dissectors/packet-pflog.c

Copyright: 2001 Mike Frantzen

License: BSD-2-clause-no-conditions

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- 
- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.  
with the distribution.
- 

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files: wiretap/erf.c  
      wiretap/erf.h

Copyright: 2003 Endace Technology Ltd, Hamilton, New Zealand.

License: BSD-3-clause-Endance

This software and documentation has been developed by Endace Technology Ltd.

along with the DAG PCI network capture cards. For further information please visit <https://www.endace.com/>.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of Endace Technology Ltd may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY ENDACE TECHNOLOGY LTD ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ENDACE TECHNOLOGY LTD BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files: wiretap/i4b\_trace.h  
Copyright: 1997, 1999 Hellmuth Michaelis  
License: BSD-2-clause

Files: epan/dissectors/pidl/rfr/rfr.idl  
      epan/dissectors/pidl/mapi/mapi.idl  
Copyright: 2008 Julien Kerihuel, OpenChange Project  
Comment: Note that those are interface definitions thus are not copyrightable  
      in FSF's view  
      <http://lkml.iu.edu/hypermail/linux/kernel/0301.1/0362.html>  
License: GPL-3+

Files: tools/pidl/idl.y  
Copyright: Andrew Tridgell <[tridge@samba.org](mailto:tridge@samba.org)>  
License: GPL-3+

License: BSD-2-clause  
Redistribution and use in source and binary forms, with or without modification,

are permitted provided that the following conditions  
are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: GPL-2+

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; version 2 dated June, 1991, or (at your option) any later version.

On Debian systems, the complete text of version 2 of the GNU General Public License can be found in '/usr/share/common-licenses/GPL-2'.

License: GPL-3+

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>.

On Debian systems, the full text of the GNU General Public License version 3 can be found in the file '/usr/share/common-licenses/GPL-3'.

License: ISC

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

.  
THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

License: LGPL-2+

Copyright (C) 1991, 1992, 1996, 1998 Free Software Foundation, Inc.  
This file is part of the GNU C Library.

.  
The GNU C Library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

.  
The GNU C Library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

.  
You should have received a copy of the GNU Library General Public License along with the GNU C Library; see the file COPYING.LIB. If not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA.

All files in this directory are dual licensed by the licenses included below, unless the individual license of a file says otherwise.

```
#####  
#  
# Copyright (c) ....  
#  
# This program is free software; you can redistribute it and/or  
# modify it under the terms of the GNU General Public License  
# as published by the Free Software Foundation; either version 2  
# of the License, or (at your option) any later version.  
#
```

```

# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 51 Franklin Street, Fifth Floor,
# Boston, MA 02110-1301, USA.
#
#####
#
#
    Copyright (c) ...
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions
# are met:
#
# * Redistributions of source code must retain the above copyright
# notice, this list of conditions and the following disclaimer.
#
# * Redistributions in binary form must reproduce the above copyright
# notice, this list of conditions and the following disclaimer in
# the documentation and/or other materials provided with the
# distribution.
#
# * Neither the name of the <ORGANIZATION> nor the names of its
# contributors may be used to endorse or promote products derived
# from this software without specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
    IN NO EVENT SHALL THE
# COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
# INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
# BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
# LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
# CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
# LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
# ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
# POSSIBILITY OF SUCH DAMAGE.
#
The IDL files in this directory are made available by the Samba Team
under the following license:

```

Permission to use, copy, modify, and distribute these interface definitions for any purpose is hereby granted without fee.

This work is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

## 1.50 sysstat 12.7.6-r1

### 1.50.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc., <<http://fsf.org/>>  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the

source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you



conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program

except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is

implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED

OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
{description}  
Copyright (C) {year} {fullname}
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by

the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type
`show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
{signature of Ty Coon}, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.  
The MIT License (MIT)

Copyright (c) 2015 Lance W. Shelton

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.51 libpcap 1.10.5-r1

### 1.51.1 Available under license :

License: BSD

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

## 1.52 jaraco.collections 5.1.0

### 1.52.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.53 libbpf 1.5.1-r0

### 1.53.1 Available under license :

Valid-License-Identifier: BSD-2-Clause

SPDX-URL: <https://spdx.org/licenses/BSD-2-Clause.html>

Usage-Guide:

To use the BSD 2-clause "Simplified" License put the following SPDX tag/value pair into a comment according to the placement guidelines in the licensing rules documentation:

SPDX-License-Identifier: BSD-2-Clause

License-Text:

Copyright (c) 2015 The Libbpf Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LGPL-2.1 OR BSD-2-Clause



# 1.54 libgpg-error 1.55-r0

## 1.54.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in

whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work

based on it,  
under Section 2) in object code or executable form under the terms of  
Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6.

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING

WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),

EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.

The hypothetical commands

`show w' and `show c' should show the appropriate  
parts of the General Public License. Of course, the commands you use may  
be called something other than `show w' and `show c'; they could even be  
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your  
school, if any, to sign a "copyright disclaimer" for the program, if  
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into  
proprietary programs. If your program is a subroutine library, you may  
consider it more useful to permit linking proprietary applications with the  
library. If this is what you want to do, use the GNU Library General  
Public License instead of this License.

## GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts  
as the successor of the GNU Library Public License, version 2, hence  
the version number 2.1.]

### Preamble

The licenses for most software are designed to take away your  
freedom to share and change it. By contrast, the GNU General Public  
Licenses are intended to guarantee your freedom to share and change  
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some  
specially designated software packages--typically libraries--of the  
Free Software Foundation and other authors who decide



to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the

ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must

be combined with the library in order to run.

## GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or

collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary

GNU General Public

License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses

the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood

that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

- b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to



refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free

Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

##### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990

Ty Coon, President of Vice

That's all there is to it!

#### LICENSE TERMS

Copyright (c) 2000, Dimitrios Souflis  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Dimitrios Souflis nor the names of the contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.55 ca-certificates 20250911-r0

### 1.55.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/opt/cola/permits/2660733757_1759915527.765318/0/ca-certificates-20250911-r0-tar-gz/usr/share/ca-
certificates/mozilla/HARICA_TLS_RSA_Root_CA_2021.crt-----BEGIN CERTIFICATE-----
/opt/cola/permits/2660733757_1759915527.765318/0/ca-certificates-20250911-r0-tar-gz/usr/share/ca-
certificates/mozilla/QuoVadis_Root_CA_2.crt-----BEGIN CERTIFICATE-----
/opt/cola/permits/2660733757_1759915527.765318/0/ca-certificates-20250911-r0-tar-gz/usr/share/ca-
certificates/mozilla/TWCA_Global_Root_CA.crt-----BEGIN CERTIFICATE-----
/opt/cola/permits/2660733757_1759915527.765318/0/ca-certificates-20250911-r0-tar-gz/usr/share/ca-
certificates/mozilla/TrustAsia_TLS_RSA_Root_CA.crt-----BEGIN CERTIFICATE-----
```

Found in path(s):

\*

No license file was found, but licenses were detected in source scan.

```
-----BEGIN CERTIFICATE-----
MIIFWjCCA0KgAwIBAgISEdK7udcjGJ5AXwqdLdDfJWfRMA0GCSqGSIb3DQEBAUA
MEYx CzAJBgNVBAYTAkJFMRkwFwYDVQQKEExBHbG9iYWxTaWduIG52LXNhMRwwGgYD
VQQDExNHbG9iYWxTaWduIFJvb3QgUjQ2MB4XDTE5MDMyMDAwMDAwMFoXDTQ2MDMy
MDAwMDAwMFowRjELMAkGA1UEBhMCQkUxGTAXBgNVBAoTEEdsb2JhbFNpZ24gbnYt
c2ExHDAaBgNVBAMTE0dsb2JhbFNpZ24gUm9vdCBSNDYwgglhMA0GCSqGSIb3DQEB
AQUAA4ICDwAwggIKAoICAQCsrHQy6LNI5brtQyYdpokNRbopiLKkHWPd08EsCvEJ
OaFV6Wc0dwXu5FUdUiXSE2te4R2pt32JmI8Nnp8semNgQB+msLZ4j5lUlgHYruQG
vGIFaHa/r6gjA7aUD7xubMLL1aa7DOn2wQL7Id5m3RerdELv8HQvJfTqa1VbkNud
316HCkd7rRlr+/fKYIje2sGP1q7Vf9Q8g+7XFkyDRTNrJ9CG0Bwta/OrffGFqfUo
0q3v84RLHif8E6M6cqJaESvWJ3En7YEtbWaBkoe0G1h6zD8K+kZPTXhc+CtI4wSE
```

y132tGqzZfxCnlEmIyDLPRT5ge1lFgBPGmSXZgjPjHvjK8Cd+RTyG/FWaha/LIWF  
zXg4mutCagI0GIMXTpRW+LaCtfOW3T3zvn8gdz57GSNrLNRyc0NXfeD412lPFzYE  
+cCQYDdF3uYM2HSNrpyibXRdQr4G9dlkbgIQRImwTDsHTUB+JMWKmlJ5jqSngiCN  
I/onccnfxkF0oE32kRbcRoxfKWMxWxEM2G/CtjJ9++ZdU6Z+Ffy7dXxd7Pj2Fxs  
x2sZy/N78CsHpdlseVR2bJ0cpm4O6XkMqCNqo98bMDGfsVR7/mrLZqrcZdCinkqa  
ByFrgY/bxFn63iLABJzjqls2k+g9vXqhnQt2sQvHnf3PmKgGwvqqo6GDoLclcqUC  
4wIDAQABo0IwQDAOBgNVHQ8BAf8EBAMCAYYwDwYDVROTAQH/BAUwAwEB/zAdBgNV  
HQ4EFgQUA1yrc4GHqMywptWU4jaWSf8FmSwwDQYJKoZIhvcNAQEMBQADggIBAHx4  
7PYCLLtbfPirXTncvtgdokIzTfnvpCo7RGkerNlFo048p9gkUbJUHJNOxO97k4Vg  
JuoJSOD1u8fpaNK7ajFxzHmuEajwmf3IH7wvqMxX63bElazHU1VNaL8FpO7XJqti  
2kM3S+LGteWygxk6x9PbTZ4IevPuzz5i+6zoYMzRx6Fc0XERczzF2sUyQQCptlk  
pnnpHs6i58FZFZ8d4kuaPp92CC1r2LpXFNqD6v6MVenQTqnMdzGxRBF6XLE+0xRF  
FRhiJBPSy03OXIPBNvIQ6IbbjhVp+J3pZmOUdkLG5NrmJ7v2B0GbhWrJKsFjLt  
rWhV/pi60zTe9Mlhww6G9kuEYO4Ne7UyWHmRVSyBQ7N0H3qqJZ4d16GLuc1CLgSk  
ZoNNiTW2bKg2SnkheCLQqRzRQDGQob4Ez8pn7fXwgNNgyYMqIgXQBztSvwyeviv5  
u+YfjyW6hY0XHgL+XVAEV8/+LbZvXMAaq7afJMbfc2hIkCwU9D9SGuTSyXTDYWnP  
4vkYxboznxSjBF25cfe1lNj2M8FawTSLfJvdkzrnE6JwYZ+vY+XX4M2bUdGc6  
N3ec592kD3ZDZopD8p/7DEJ4Y9HiD2971KE9dJeFt0g5QdYg/NA6s/rob8SKunE3  
vouXsXgxT7PntgMTzlSdriVZzH81Xwj3QEUxCp6  
-----END  
CERTIFICATE-----

Found in path(s):

\*

/ca-certificates-20250911-r0-tar-gz/usr/share/ca-certificates/mozilla/GlobalSign\_Root\_R46.crt

## 1.56 libidn 2.3.7-r0

### 1.56.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates  
the terms and conditions of version 3 of the GNU General Public  
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser  
General Public License, and the "GNU GPL" refers to version 3 of the GNU  
General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

#### 1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

#### 2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

- a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or
- b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

#### 3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from

a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

- a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the object code with a copy of the GNU GPL and this license document.

#### 4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

- a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the Combined Work with a copy of the GNU GPL and this license document.
- c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.
- d) Do one of the following:

- 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

- 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer

system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

## 5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
- b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

## 6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you



received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

## GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License

along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of

a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under

this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to

decide if he or she is willing  
to distribute software through any other system and a licensee cannot  
impose that choice.

This section is intended to make thoroughly clear what is believed to  
be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in  
certain countries either by patents or by copyrighted interfaces, the  
original copyright holder who places the Program under this License  
may add an explicit geographical distribution limitation excluding  
those countries, so that distribution is permitted only in or among  
countries not thus excluded. In such case, this License incorporates  
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions  
of the General Public License from time to time. Such new versions will  
be similar in spirit to the present version, but may differ in detail to  
address new problems or concerns.

Each version is given a distinguishing  
version number. If the Program  
specifies a version number of this License which applies to it and "any  
later version", you have the option of following the terms and conditions  
either of that version or of any later version published by the Free  
Software Foundation. If the Program does not specify a version number of  
this License, you may choose any version ever published by the Free Software  
Foundation.

10. If you wish to incorporate parts of the Program into other free  
programs whose distribution conditions are different, write to the author  
to ask for permission. For software which is copyrighted by the Free  
Software Foundation, write to the Free Software Foundation; we sometimes  
make exceptions for this. Our decision will be guided by the two goals  
of preserving the free status of all derivatives of our free software and  
of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE,  
THERE IS NO WARRANTY  
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN  
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES  
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED  
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF  
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS  
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE  
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,

## REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute  
it  
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this

is what you want to do, use the GNU Lesser General  
Public License instead of this License.

Libidn2 COPYING -- Licensing information.                    -\*- outline -\*-  
Copyright (C) 2011-2016 Simon Josefsson  
See the end for copying conditions.

The source code for the C library (libidn2.a or libidn.so) are licensed under the terms of either the GNU General Public License version 2.0 or later (see the file COPYINGv2) or the GNU Lesser General Public License version 3.0 or later (see the file COPYING.LESSERv3), or both in parallel as here.

The command line tool, self tests, examples, and other auxiliary files, are licensed under the GNU General Public License version 3.0 or later.

The license of the Unicode character data files (which are parsed into static storage in the library) are documented in COPYING.unicode.

Other files are licensed as indicated in each file.

There may be exceptions to these general rules, see each file for



precise information.

-----  
This file is free software: you can redistribute  
it and/or modify it  
under the terms of the GNU General Public License as published by the  
Free Software Foundation, either version 3 of the License, or (at your  
option) any later version.

This file is distributed in the hope that it will be useful, but  
WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU  
General Public License for more details.

You should have received a copy of the GNU General Public License  
along with this file. If not, see <<http://www.gnu.org/licenses/>>.

#### A. Unicode Copyright.

Copyright 1991-2016 Unicode, Inc. All rights reserved.

Certain documents and files on this website contain a legend indicating that "Modification is permitted." Any person is hereby authorized, without fee, to modify such documents and files to create derivative works conforming to the Unicode Standard, subject to Terms and Conditions herein.

Any person is hereby authorized, without fee, to view, use, reproduce, and distribute all documents and files solely for informational purposes and in the creation of products supporting the Unicode Standard, subject to the Terms and Conditions herein.

Further specifications of rights and restrictions pertaining to the use of the particular set of data files known as the "Unicode Character Database" can be found in the License.

Each version of the Unicode Standard has further specifications of rights and restrictions of use. For the book editions (Unicode 5.0 and earlier), these are found on the back of the title page. The online code charts carry specific restrictions. All other files, including online documentation of the core specification for Unicode 6.0 and later, are covered under these general Terms of Use.

No license is granted to "mirror" the Unicode website where a fee is charged for access to the "mirror" site.

Modification is not permitted with respect to this document. All copies of this document must be verbatim.

#### B. Restricted Rights Legend.

Any technical data or software which is licensed to the United States of America, its agencies and/or instrumentalities under this Agreement is commercial technical data or commercial computer software developed exclusively at private expense as defined in FAR 2.101, or DFARS 252.227-7014 (June 1995), as applicable. For technical data, use, duplication, or disclosure by the Government is subject to restrictions as set forth in DFARS 202.227-7015 Technical Data, Commercial and Items (Nov 1995) and this Agreement. For Software, in accordance with FAR 12-212 or DFARS 227-7202, as applicable, use, duplication or disclosure by the Government is subject to the restrictions set forth in this Agreement.

#### C. Warranties and Disclaimers.

This publication and/or website may include technical or typographical errors or other inaccuracies. Changes are periodically added to the information herein; these changes will be incorporated in new editions of the publication and/or website. Unicode may make improvements and/or changes in the product(s) and/or program(s) described in this publication and/or website at any time.

If this file has been purchased on magnetic or optical media from Unicode, Inc. the sole and exclusive remedy for any claim will be exchange of the defective media within ninety (90) days of original purchase.

EXCEPT AS PROVIDED IN SECTION C.2, THIS PUBLICATION AND/OR SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND EITHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO,

ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. UNICODE AND ITS LICENSORS ASSUME NO RESPONSIBILITY FOR ERRORS OR OMISSIONS IN THIS PUBLICATION AND/OR SOFTWARE OR OTHER DOCUMENTS WHICH ARE REFERENCED BY OR LINKED TO THIS PUBLICATION OR THE UNICODE WEBSITE.

#### D. Waiver of Damages.

In no event shall Unicode or its licensors be liable for any special, incidental, indirect or consequential damages of any kind, or any damages whatsoever, whether or not Unicode was advised of the possibility of the damage, including, without limitation, those resulting from the following: loss of use, data or profits, in connection with the use, modification or distribution of this information or its derivatives.

#### E. Trademarks & Logos.

The Unicode Word Mark and the Unicode Logo are trademarks of Unicode, Inc. The Unicode Consortium and Unicode, Inc. are trade names of Unicode, Inc. Use of the information and materials found on this website indicates your acknowledgement

of Unicode, Inc.'s exclusive worldwide rights in the Unicode Word Mark, the Unicode Logo, and the Unicode trade names.

The Unicode Consortium Name and Trademark Usage Policy (Trademark Policy) are incorporated herein by reference and you agree to abide by the provisions of the Trademark Policy, which may be changed from time to time in the sole discretion of Unicode, Inc.

All third party trademarks referenced herein are the property of their respective owners.

#### F. Miscellaneous.

**Jurisdiction and Venue.** This server is operated from a location in the State of California, United States of America. Unicode makes no representation that the materials are appropriate for use in other locations. If you access this server from other locations, you are responsible for compliance with local laws. This Agreement, all use of this site and any claims and damages resulting from use of this site are governed solely by the laws of the State of California without regard to any principles

which would apply the laws of a different jurisdiction. The user agrees that any disputes regarding this site shall be resolved solely in the courts located in Santa Clara County, California. The user agrees said courts have personal jurisdiction and agree to waive any right to transfer the dispute to any other forum.

**Modification by Unicode** Unicode shall have the right to modify this Agreement at any time by posting it to this site. The user may not assign any part of this Agreement without Unicode's prior written consent.

**Taxes.** The user agrees to pay any taxes arising from access to this website or use of the information herein, except for those based on Unicode's net income.

**Severability.** If any provision of this Agreement is declared invalid or unenforceable, the remaining provisions of this Agreement shall remain in effect.

**Entire Agreement.** This Agreement constitutes the entire agreement between the parties.

#### EXHIBIT 1

Unicode Data Files include all data files

under the directories  
<http://www.unicode.org/Public/>, <http://www.unicode.org/reports/>,  
<http://www.unicode.org/cldr/data/>, <http://source.icu-project.org/repos/icu/>, and  
<http://www.unicode.org/utility/trac/browser/>.

Unicode Data Files do not include PDF online code charts under the  
directory <http://www.unicode.org/Public/>.

Software includes any source code published in the Unicode Standard  
or under the directories  
<http://www.unicode.org/Public/>, <http://www.unicode.org/reports/>,  
<http://www.unicode.org/cldr/data/>, <http://source.icu-project.org/repos/icu/>, and  
<http://www.unicode.org/utility/trac/browser/>.

NOTICE TO USER: Carefully read the following legal agreement.  
BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.'S  
DATA FILES ("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"),  
YOU UNEQUIVOCALLY ACCEPT, AND AGREE TO BE BOUND BY, ALL OF THE  
TERMS AND CONDITIONS OF THIS AGREEMENT.  
IF YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE  
THE DATA FILES OR SOFTWARE.

#### COPYRIGHT AND PERMISSION NOTICE

Copyright 1991-2016 Unicode, Inc. All rights reserved.  
Distributed under the Terms of Use in <http://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining  
a copy of the Unicode data files and any associated documentation  
(the "Data Files") or Unicode software and any associated documentation  
(the "Software") to deal in the Data Files or Software  
without restriction, including without limitation the rights to use,  
copy, modify, merge, publish, distribute, and/or sell copies of  
the Data Files or Software, and to permit persons to whom the Data Files  
or Software are furnished to do so, provided that either  
(a) this copyright and permission notice appear with all copies  
of the Data Files or Software, or  
(b) this copyright and permission notice appear in associated  
Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF  
ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED  
TO THE  
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND  
NONINFRINGEMENT OF THIRD PARTY RIGHTS.  
IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS  
NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL

DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

## 1.57 wheel 0.45.1

### 1.57.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only

on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

#### END OF TERMS AND CONDITIONS

This software is made available under the terms of *\*either\** of the licenses found in LICENSE.APACHE or LICENSE.BSD. Contributions to this software is made under the terms of *\*both\** these licenses.

MIT License

Copyright (c) 2012 Daniel Holth <dholth@fastmail.fm> and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING

FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) Donald Stufft and individual contributors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE



DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.58 Iz4 1.10.0-r0

### 1.58.1 Available under license :

This repository uses 2 different licenses :

- all files in the `lib` directory use a BSD 2-Clause license
- all other files use a GPL-2.0-or-later license, unless explicitly stated otherwise

Relevant license is reminded at the top of each source file,  
and with presence of COPYING or LICENSE file in associated directories.

This model is selected to emphasize that  
files in the `lib` directory are designed to be included into 3rd party applications,  
while all other files, in `programs`, `tests` or `examples`,  
are intended to be used "as is", as part of their intended scenarios,  
with no intention to support 3rd party integration use cases.  
Copyright (c) 2014, lpsantil  
All rights reserved.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions are met:

\* Redistributions of source code must retain the above copyright notice, this  
list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice,  
this list of conditions and the following disclaimer in the documentation  
and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"  
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE  
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE  
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL  
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR  
SERVICES;

LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER  
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,  
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE

OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Format: <http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Upstream-Name: liblz4

Upstream-Contact: Yann Collet <Cyan4973@github.com>

Source: <https://github.com/lz4/lz4>

Files: \*

Copyright: (C) 2011-2020 Yann Collet

License: GPL-2+

The full text of license: <https://github.com/lz4/lz4/blob/dev/lib/LICENSE>

LZ4 Library

Copyright (c) 2011-2020, Yann Collet

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

\* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Full name:

GNU General Public License v2.0 or later

Short identifier:

GPL-2.0-or-later

Text:

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work

which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its

scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies

of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and

its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of

Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding

source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work

for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of

that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version

published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

# 1.59 mtr 0.96-r0

## 1.59.1 Available under license :

The Wolfi build recipes themselves are provided under the Apache License 2.0, which is included for your convenience below.

The underlying packages are provided under the license terms provided by each FOSS project supplying the packages.

Any image built from Wolfi (or any other distribution) is provided under the combination of license terms resulting from the use of individual packages; we recommend inspecting the SBOMs generated by apko when building Wolfi images to understand the license terms of any individual image, or using an SCA tool such as Syft.

-----  
Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,



worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties

or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even

if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]"

replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

MIT License

Copyright (c) 2014-2021 Ben Balter and Licensee contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

package:

name: py3-license-expression

version: "30.4.4"

epoch: 1

description: license-expression is a comprehensive utility library to parse, compare, simplify and normalize license expressions (such as SPDX license expressions) using boolean logic.

copyright:

- license: Apache-2.0

dependencies:

provider-priority: 0

vars:

pypi-package: license-expression

import: license\_expression

data:

- name: py-versions



items:

3.10: '310'  
3.11: '311'  
3.12: '312'  
3.13: '313'

environment:

contents:

packages:

- py3-supported-build-base

pipeline:

- uses: git-checkout

with:

repository: <https://github.com/nexB/license-expression>

tag: v\${{package.version}}

expected-commit: a3c00c09986bcca1240afd5b1844b56de3f581c1

subpackages:

- range: py-versions

name: py\${{range.key}}-\${{vars.pypi-package}}

description: python\${{range.key}} version of \${{vars.pypi-package}}

dependencies:

provider-priority: \${{range.value}}

provides:

- \${{package.name}}

- py3-\${{vars.pypi-package}}

runtime:

- py\${{range.key}}-boolean.py

pipeline:

- uses: py/pip-build-install

with:

python: python\${{range.key}}

- uses: strip

test:

pipeline:

- uses: python/import

with:

python: python\${{range.key}}

imports: |

import \${{vars.import}}

- name: py3-supported-\${{vars.pypi-package}}

description: meta package providing \${{vars.pypi-package}} for supported python versions.

dependencies:

runtime:

- py3.10-\${{vars.pypi-package}}

- py3.11-\${{vars.pypi-package}}

- py3.12-`{{vars.pypi-package}}`
- py3.13-`{{vars.pypi-package}}`

test:

pipeline:

- uses: test/metapackage

test:

pipeline:

- uses: python/import

with:

imports: |

import `{{vars.import}}`

update:

enabled: true

github:

identifier: nexB/license-expression

strip-prefix: v

package:

name: libart-lgpl

version: 2.3.21

epoch: 2

description: A library for high-performance 2D graphics

copyright:

- license: LGPL-2.0-or-later

environment:

contents:

packages:

- autoconf
- automake
- build-base
- busybox
- ca-certificates-bundle
- libtool

pipeline:

- uses: fetch

with:

expected-sha256: fdc11e74c10fc9ffe4188537e2b370c0abacca7d89021d4d303afdf7fd7476fa

uri: <https://download.gnome.org/sources/libart-lgpl/2.3/libart-lgpl-{{package.version}}.tar.bz2>

- runs: |
  - libtoolize --force && aclocal && autoconf --force && automake --add-missing
- runs: autoreconf -vif
- uses: autoconf/configure

```

with:
  opts: |
    --disable-static

- uses: autoconf/make

- uses: autoconf/make-install

- uses: strip

subpackages:
- name: libart-lgpl-dev
  pipeline:
    - uses: split/dev
  dependencies:
    runtime:
      - libart-lgpl
  description: libart-lgpl dev
  test:

  pipeline:
    - runs: |
        libart2-config --version
    - uses: test/pkgconf
    - uses: test/tw/ldd-check

update:
  enabled: true
  release-monitor:
    identifier: 21864

test:
  pipeline:
    - uses: test/tw/ldd-check
package:
  name: go-licenses
  version: "2.0.1"
  epoch: 2 # CVE-2025-61729
  description: A lightweight tool to report on the licenses used by a Go package and its dependencies. Highlight!
  Versioned external URL to licenses can be found at the same time.
  copyright:
    - license: Apache-2.0

  pipeline:
    - uses: git-checkout
    with:
      repository: https://github.com/google/go-licenses
      tag: v${{package.version}}

```

expected-commit: 3e084b0caf710f7bfead967567539214f598c0a2

- uses: go/build

with:

modroot: .

packages: .

output: go-licenses

update:

enabled: true

github:

identifier: google/go-licenses

strip-prefix: v

test:

pipeline:

# AUTOGENERATED

- runs: |

go-licenses --help

issuer: https://accounts.google.com

# staging-enforce: ci-license-checka7daalc99klekq@staging-enforce-cd1e.iam.gserviceaccount.com  
(111508570611001124410)

# prod-images: TODO

subject\_pattern: "(111508570611001124410)"

permissions:

checks: write

pull\_requests: write # need to comment on PRs

# 1.60 iptables 1.8.11-r1

## 1.60.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your  
freedom to share and change it. By contrast, the GNU General Public  
License is intended to guarantee your freedom to share and change free  
software--to make sure the software is free for all its users. This

General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a

notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such

an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues),



conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any

later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively

convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into

proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

## 1.61 ipvsadm 1.31-r3

### 1.61.1 Available under license :

-----

ipvsadm - Version 1.14 - 11th January 2001

This is free software. See below for details.

ipvsadm is a utility to administer the IP virtual server services offered by the Linux kernel with IP virtual server support.

This version of ipvsadm requires IPVS module v0.0.5 for kernel 2.4 or later. Check out the Linux Virtual Server Project home page on the World Wide Web:

<http://www.LinuxVirtualServer.org/>

or

<http://www.Linux-VS.org/>

for the most recent information and original sources about ipvsadm.

To make, make sure your Linux kernel is already patched with IPVS and "make menuconfig" or "make xconfig" to setup the right compiling options, see the README of the virtual server patch for detail, then simply type make in the source directory.

Install to your liking. We suggest the following pathnames:

/sbin/ipvsadm

/usr/man/man8/ipvsadm.8

This will be done automatically when calling

make install

in the

source directory.

Wensong Zhang <[wensong@linuxvirtualserver.org](mailto:wensong@linuxvirtualserver.org)>

-----

This program is free software; you can redistribute it and/or modify

it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

-----

## 1.62 fping 5.3-r0

### 1.62.1 Available under license :

- \* Original author: Roland Schemers <schemers@stanford.edu>
- \* IPv6 Support: Jeroen Massar <jeroen@unfix.org / jeroen@ipng.nl>
- \* Improved main loop: David Schweikert <david@schweikert.ch>
- \* Debian Merge, TOS settings: Tobi Oetiker <tobi@oetiker.ch>
- \* Bugfixes, byte order & senseful seq.-numbers: Stephan Fuhrmann (stephan.fuhrmann AT 1und1.de)
- \*
- \* Redistribution and use in source and binary forms are permitted
- \* provided that the above copyright notice and this paragraph are
- \* duplicated in all such forms and that any documentation,
- \* advertising materials, and other materials related to such
- \* distribution and use acknowledge that the software was developed
- \* by Stanford University. The name of the University may not be used
- \* to endorse or promote products derived from this software without
- \* specific prior written permission.
- \* THIS SOFTWARE IS PROVIDED ``AS IS" AND WITHOUT ANY EXPRESS OR
- \* IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
- \* WARRANTIES
- OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

## 1.63 bash 5.2.37-r0

### 1.63.1 Available under license :

Shellfloat is copyright (c) 2020 by Michael Wood.

=====

GNU GENERAL PUBLIC LICENSE  
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<https://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code.

And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed

this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS

### 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the

public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

#### 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for



the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

## 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met.

This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

## 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid

circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

#### 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

#### 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

## 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the

Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no

charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply

if neither you nor  
any third party retains the ability to install  
modified object code on the User Product (for example, the work has  
been installed in ROM).

The requirement to provide Installation Information does not include a  
requirement to continue to provide support service, warranty, or updates  
for a work that has been modified or installed by the recipient, or for  
the User Product in which it has been modified or installed. Access to a  
network may be denied when the modification itself materially and  
adversely affects the operation of the network or violates the rules and  
protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided,  
in accord with this section must be in a format that is publicly  
documented (and with an implementation available to the public in  
source code form), and must require no special password or key for  
unpacking, reading or copying.

## 7. Additional Terms.

"Additional permissions" are terms that supplement  
the terms of this

License by making exceptions from one or more of its conditions.  
Additional permissions that are applicable to the entire Program shall  
be treated as though they were included in this License, to the extent  
that they are valid under applicable law. If additional permissions  
apply only to part of the Program, that part may be used separately  
under those permissions, but the entire Program remains governed by  
this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option  
remove any additional permissions from that copy, or from any part of  
it. (Additional permissions may be written to require their own  
removal in certain cases when you modify the work.) You may place  
additional permissions on material, added by you to a covered work,  
for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you  
add to a covered work, you may (if  
authorized by the copyright holders of  
that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the  
terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or  
author attributions in that material or in the Appropriate Legal

Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

## 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your

license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation

by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

#### 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept

this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

#### 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor

has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

#### 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent



license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

## 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

### 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

### 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number.

If the

Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

### 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY

APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF

ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of

MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>  
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <https://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <https://www.gnu.org/licenses/why-not-lgpl.html>.

GNU Free Documentation License  
Version 1.3, 3 November 2008

Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.  
<http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it,

with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We

have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of

the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The

Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition.

Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated

as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

#### 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.



- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains

nothing but

endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections

Entitled "History"

in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4.

Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include

the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

## 11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of

the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

GNU GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed

to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it,

either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies  
of the Program's

source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)



These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to

control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent

license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.

#### EXCEPT WHEN

OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN

IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

@c The GNU Free Documentation License.

@center Version 1.3, 3 November 2008

@c This file is intended to be included within another document,  
@c hence no sectioning command or @node.

@display

Copyright @copyright{} 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

@uref{http://fsf.org/}

Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

@end display

@enumerate 0

@item

PREAMBLE

The purpose of this License is to make a manual, textbook, or other  
functional and useful document @dfn{free} in the sense of freedom: to  
assure everyone the effective freedom to copy and redistribute it,  
with or without modifying it, either commercially or noncommercially.  
Secondarily, this License preserves for the author and publisher a way  
to get credit for their work, while not being considered responsible  
for modifications made by others.

This License is a kind of ``copyleft'', which means that derivative  
works of the document  
must themselves be free in the same sense. It  
complements the GNU General Public License, which is a copyleft  
license designed for free software.

We have designed this License in order to use it for manuals for free  
software, because free software needs free documentation: a free  
program should come with manuals providing the same freedoms that the  
software does. But this License is not limited to software manuals;  
it can be used for any textual work, regardless of subject matter or  
whether it is published as a printed book. We recommend this License  
principally for works whose purpose is instruction or reference.

@item

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that  
contains a notice placed by the copyright holder saying it can be  
distributed under the terms of this License. Such a notice grants a  
world-wide, royalty-free license, unlimited in duration, to use that  
work under the conditions stated herein. The ``Document'', below,  
refers

to any such manual or work. Any member of the public is a licensee, and is addressed as ``you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A ``Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A ``Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The ``Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The ``Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A ``Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not ``Transparent" is called ``Opaque".

Examples of suitable formats for Transparent copies include plain `@sc{ascii}` without markup, Texinfo input format, `La@TeX{}` input format, `@acronym{SGML}` or `@acronym{XML}` using a publicly available `@acronym{DTD}`, and standard-conforming simple `@acronym{HTML}`, PostScript or `@acronym{PDF}` designed for human modification. Examples of transparent image formats include `@acronym{PNG}`, `@acronym{XCF}` and `@acronym{JPG}`. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, `@acronym{SGML}` or `@acronym{XML}` for which the `@acronym{DTD}` and/or processing tools are not generally available, and the machine-generated `@acronym{HTML}`, PostScript or `@acronym{PDF}` produced by some word processors for output purposes only.

The ``Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, ``Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The ``publisher" means any person or entity that distributes copies of the Document to the public.

A section ``Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as ``Acknowledgements", ``Dedications", ``Endorsements", or ``History".) To ``Preserve the Title" of such a section when you modify the Document means that it remains a section ``Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

@item

## VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever



to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

@item

## COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible.

You may add other material on the covers in addition.

Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that

this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

@item

## MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

@enumerate A

@item

Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

@item

List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

@item

State on the Title page the name of the publisher of the Modified Version, as the publisher.

@item

Preserve all the copyright notices of the Document.

@item

Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

@item

Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

@item

Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

@item

Include an unaltered copy of this License.

@item

Preserve the section Entitled ``History'', Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled ``History'' in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

@item

Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the ``History'' section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

@item

For any section Entitled ``Acknowledgements'' or ``Dedications'', Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

@item

Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

@item

Delete any section Entitled ``Endorsements''. Such a section may not be included in the Modified Version.

@item

Do not retitle any existing section to be Entitled ``Endorsements'' or to conflict in title with any Invariant Section.

@item

Preserve any Warranty Disclaimers.

@end enumerate

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document,

you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled ``Endorsements'', provided it contains nothing but endorsements of your Modified Version by various parties---for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

@item

## COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled ``History'' in the various original documents, forming one section Entitled

``History"; likewise combine any sections Entitled ``Acknowledgements", and any sections Entitled ``Dedications". You must delete all sections Entitled ``Endorsements."

@item

## COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract

a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

@item

## AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an ``aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

@item

## TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the

original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled ``Acknowledgements'', ``Dedications'', or ``History'', the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

@item

## TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

@item

## FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new

versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See [@uref{http://www.gnu.org/copyleft/}](http://www.gnu.org/copyleft/).

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License ``or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

@item  
RELICENSING

``Massive Multiauthor Collaboration Site" (or ``MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A ``Massive Multiauthor Collaboration" (or ``MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

``CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

``Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is ``eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the

MMC is eligible for relicensing.

@end enumerate

@page

@heading ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

@smallexample

@group

Copyright (C) @var{year} @var{your name}.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled ``GNU Free Documentation License".

@end group

@end smallexample

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the ``with@dots{ } Texts." line with this:

@smallexample

@group

with the Invariant Sections being @var{list their titles}, with the Front-Cover Texts being @var{list}, and with the Back-Cover Texts being @var{list}.

@end group

@end smallexample

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

@c Local Variables:

@c ispell-local-pdict: "ispell-dict"

@c End:



## GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### Preamble

The GNU General Public License is a free, copyleft license for  
software and other kinds of works.

The licenses for most software and other practical works are designed  
to take away your freedom to share and change the works. By contrast,  
the GNU General Public License is intended to guarantee your freedom to  
share and change all versions of a program--to make sure it remains free  
software for all its users. We, the Free Software Foundation, use the  
GNU General Public License for most of our software; it applies also to  
any other work released this way by its authors. You can apply it to  
your programs, too.

When we speak of free software, we are referring to  
freedom, not  
price. Our General Public Licenses are designed to make sure that you  
have the freedom to distribute copies of free software (and charge for  
them if you wish), that you receive source code or can get it if you  
want it, that you can change the software or use pieces of it in new  
free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you  
these rights or asking you to surrender the rights. Therefore, you have  
certain responsibilities if you distribute copies of the software, or if  
you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether  
gratis or for a fee, you must pass on to the recipients the same  
freedoms that you received. You must make sure that they, too, receive  
or can get the source code. And you must show them these terms so they  
know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1)

assert copyright on the software, and (2) offer you this License  
giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains  
that there is no warranty for this free software. For both users' and  
authors' sake, the GPL requires that modified versions be marked as

changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS

### 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without

permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

#### 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The

"System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally

available free

programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

## 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered

by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed;

section 10

makes it unnecessary.

## 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or

similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

#### 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

#### 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

## 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated

place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the

User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

## 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or



limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on

those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

## 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or

modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

#### 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

#### 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that

transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights

granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

#### 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this

License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

## 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may

not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

### 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

### 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

### 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS"

WITHOUT WARRANTY

OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17.

Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by

the Free Software Foundation, either version 3 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.

You should have received a copy of the GNU General Public License  
along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short  
notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>  
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate  
parts of the General Public License. Of course, your program's commands  
might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school,  
if any, to sign a "copyright disclaimer" for the program, if necessary.  
For more information on this, and how to apply and follow the GNU GPL, see  
<<http://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program  
into proprietary programs. If your program is a subroutine library, you  
may consider it more useful to permit linking proprietary applications with  
the library. If this is what you want to  
do, use the GNU Lesser General  
Public License instead of this License. But first, please read  
<<http://www.gnu.org/philosophy/why-not-lgpl.html>>.  
Unless otherwise stated, all files in this directory are Copyright (C)  
1991,1992,1993,1994,1995,1996,1997,1998,1999,2000,2001,2002,2003,  
2004,2005,2006,2007,2008,2009,2010,2011  
Free Software Foundation, Inc.

See the file COPYING in the bash distribution root directory for copying  
and usage restrictions.

The file ifs-posix.tests is Copyright (C) 2005 Glen Fowler.

# 1.64 ethtool 6.14.1-r0

## 1.64.1 Available under license :

ethtool is available under the terms of the GNU Public License version 2.

See COPYING for details.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and



(2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under

the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such

parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to

decide if he or she is willing  
to distribute software through any other system and a licensee cannot  
impose that choice.

This section is intended to make thoroughly clear what is believed to  
be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in  
certain countries either by patents or by copyrighted interfaces, the  
original copyright holder who places the Program under this License  
may add an explicit geographical distribution limitation excluding  
those countries, so that distribution is permitted only in or among  
countries not thus excluded. In such case, this License incorporates  
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions  
of the General Public License from time to time. Such new versions will  
be similar in spirit to the present version, but may differ in detail to  
address new problems or concerns.

Each version is given a distinguishing  
version number. If the Program  
specifies a version number of this License which applies to it and "any  
later version", you have the option of following the terms and conditions  
either of that version or of any later version published by the Free  
Software Foundation. If the Program does not specify a version number of  
this License, you may choose any version ever published by the Free Software  
Foundation.

10. If you wish to incorporate parts of the Program into other free  
programs whose distribution conditions are different, write to the author  
to ask for permission. For software which is copyrighted by the Free  
Software Foundation, write to the Free Software Foundation; we sometimes  
make exceptions for this. Our decision will be guided by the two goals  
of preserving the free status of all derivatives of our free software and  
of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE,  
THERE IS NO WARRANTY  
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN  
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES  
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED  
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF  
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS  
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE  
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,

## REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR

THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU

General Public License as published by  
the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along  
with this program; if not, write to the Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this  
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute  
it  
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this  
is what you want to do, use the GNU Lesser General  
Public License instead of this License.

## 1.65 my-test-package 1.0

### 1.65.1 Available under license :

MIT

## 1.66 libnet 1.3-r0

### 1.66.1 Available under license :

Copyright (c) 1998 - 2002 Mike D. Schiffman <mike@infonexus.com>  
Copyright (c) 2009 - 2013 Sam Roberts <vieuxtech@gmail.com>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES

(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Format-Specification: <http://wiki.debian.org/Proposals/CopyrightFormat>

Upstream-Author: Mike D. Schiffman <mike@infonexus.com>

Packaged-By: Domenico Andreoli <cavok@debian.org>

Packaged-Date: Mon, 18 Nov 2002 23:53:40 +0100

Original-Source-Location: <http://www.packetfactory.net/libnet/>

Files: debian/\*

Copyright: 2008, David Paleino <d.paleino@gmail.com>

2002-2008, Domenico Andreoli <cavok@debian.org>

License: BSD-2

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

.

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS

OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY



OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files: include/bpf.h

include/win32/getopt.h

include/win32/in\_system.h

sample/win32/getopt.c

Copyright: 1982-1997, The Regents of the University of California

License: BSD-4

\* This code is derived from the Stanford/CMU enet packet filter,

\* (net/enet.c) distributed as part of 4.3BSD, and code contributed

\* to Berkeley by Steven

McCanne and Van Jacobson both of Lawrence

\* Berkeley Laboratory.

\*

\* Redistribution and use in source and binary forms, with or without

\* modification, are permitted provided that the following conditions

\* are met:

\* 1. Redistributions of source code must retain the above copyright

\* notice, this list of conditions and the following disclaimer.

\* 2. Redistributions in binary form must reproduce the above copyright

\* notice, this list of conditions and the following disclaimer in the

\* documentation and/or other materials provided with the distribution.

\* 3. All advertising materials mentioning features or use of this software

\* must display the following acknowledgement:

\* This product includes software developed by the University of

\* California, Berkeley and its contributors.

\* 4. Neither the name of the University nor the names of its contributors

\* may be used to endorse or promote products derived from this software

\* without specific

prior written permission.

\*

\* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND

\* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

\* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

\* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE

\* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

\* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS

\* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

\* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT

\* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY

\* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF

\* SUCH DAMAGE.

Files: include/ifaddrlist.h

Copyright: 1997, The Regents of the University of California

License: BSD-3

\* Redistribution and use in source and binary forms, with or without

- \* modification, are permitted
- provided that: (1) source code distributions
- \* retain the above copyright notice and this paragraph in its entirety, (2)
- \* distributions including binary code include the above copyright notice and
- \* this paragraph in its entirety in the documentation or other materials
- \* provided with the distribution, and (3) all advertising materials mentioning
- \* features or use of this software display the following acknowledgement:
- \* ``This product includes software developed by the University of California,
- \* Lawrence Berkeley Laboratory and its contributors." Neither the name of
- \* the University nor the names of its contributors may be used to endorse
- \* or promote products derived from this software without specific prior
- \* written permission.
- \* THIS SOFTWARE IS PROVIDED ``AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED
- \* WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF
- \* MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Files: include/libnet/libnet-asn1.h

Copyright:

1998-2004, Mike D. Schiffman <mike@infonexus.com>

1988-1989, Carnegie Mellon University

License: BSD-2 | other

- \* Permission to use, copy, modify, and distribute this software and its
- \* documentation for any purpose and without fee is hereby granted,
- \* provided that the above copyright notice appear in all copies and that
- \* both that copyright notice and this permission notice appear in
- \* supporting documentation, and that the name of CMU not be
- \* used in advertising or publicity pertaining to distribution of the
- \* software without specific, written prior permission.
- \*
- \* CMU DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING
- \* ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL
- \* CMU BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR
- \* ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
- \* WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
- \* ARISING OUT OF OR IN
- CONNECTION WITH THE USE OR PERFORMANCE OF THIS
- \* SOFTWARE.

Files: sample/bgp4\*

sample/dns.c

sample/gre.c

sample/ip\_link.c

sample/ip\_raw.c

sample/sebek.c

sample/tftp.c

src/libnet\_build\_bgp.c

src/libnet\_build\_gre.c

Copyright: 2003-2004 Frdric Raynal <pappy@security-labs.org>

License: BSD-2

Files: sample/icmp\_redirect.c

Copyright: 1998-2004, Mike D. Schiffman <mike@infonexus.com>  
2003, Alberto Ornaghi <alor@antifork.org>

License: BSD-2

Files: sample/icmp\_timeexceed.c

src/libnet\_build\_sebek.c

src/libnet\_cq.c

Copyright: 1998-2004, Mike D. Schiffman <mike@infonexus.com>  
2003, Frdric Raynal <pappy@security-labs.org>

License: BSD-2

Files: sample/ospf\*

src/libnet\_build\_ospf.c

Copyright: 1998-2004, Mike D. Schiffman <mike@infonexus.com>  
1999-2000, Andrew Reiter <areiter@bindview.com>

License: BSD-2

Files: sample/ping\_of\_death.c

Copyright: 1998-2004, Mike D. Schiffman <mike@infonexus.com>  
1999-2001, Dug Song <dugsong@monkey.org>

License: BSD-2

Files: srct/libnet\_asn1.c

Copyright:

1998-2004, Mike D. Schiffman <mike@infonexus.com>

1988-1992, Carnegie Mellon University

1993-1996, 1998, The Regents of the University of California

License: BSD-3 | other

==> BSD-3 <==

- \* Redistribution and use in source and binary forms, with or without
- \* modification, are permitted provided that: (1) source code distributions
- \* retain the above copyright notice and this paragraph in its entirety, (2)
- \* distributions including binary code include the above copyright notice and
- \* this paragraph in its entirety in the documentation or other materials
- \* provided with the distribution, and (3) all advertising materials mentioning
- \* features or use of this software display the following acknowledgement:
- \* ``This product includes software developed by the University of California,
- \* Lawrence Berkeley Laboratory and its contributors." Neither the name of
- \* the University nor the names of its contributors may be used to endorse
- \* or promote products derived
- from this software without specific prior
- \* written permission.
- \* THIS SOFTWARE IS PROVIDED ``AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED
- \* WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF
- \* MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

==> other <==

- \* Permission to use, copy, modify, and distribute this software and its
- \* documentation for any purpose and without fee is hereby granted,
- \* provided that the above copyright notice appear in all copies and that
- \* both that copyright notice and this permission notice appear in
- \* supporting documentation, and that the name of CMU not be
- \* used in advertising or publicity pertaining to distribution of the
- \* software without specific, written prior permission.
- \*
- \* CMU DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING
- \* ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL
- \* CMU BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR
- \* ANY DAMAGES WHATSOEVER
- RESULTING FROM LOSS OF USE, DATA OR PROFITS,
- \* WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
- \* ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS
- \* SOFTWARE.

Files: src/libnet\_build\_fddi.c

src/libnet\_build\_rpc.c

src/libnet\_build\_token\_ring.c

Copyright: 1998-2004, Mike D. Schiffman <mike@infonexus.com>

Jason Damron <jsdamron@hushmail.com> <jdamron@stackheap.org>

License: BSD-2

Files: src/libnet\_build\_ipsec.c

Copyright: 1998-2004, Mike D. Schiffman <mike@infonexus.com>

2002, Jose Nazario <jose@crimelabs.net>

License: BSD-2

Files: src/libnet\_build\_link.c

src/libnet\_write.c

Copyright: 1998-2004, Mike D. Schiffman <mike@infonexus.com>

2003, Roberto Larcher <roberto.larcher@libero.it>

License: BSD-2

Files: src/libnet\_dll.c

Copyright: 2002, Roberto Larcher <roberto.larcher@libero.it>

License: BSD-2

Files: src/libnet\_link\_dlpi.c

Copyright: 1998-2004, Mike D. Schiffman <mike@infonexus.com>

1993-1997, The Regents of the

University of California

Atanu Ghosh <atanu@cs.ucl.ac.uk>, University College London

License: BSD-3

Files: src/libnet\_link\_nit.c

src/libnet\_link\_none.c

src/libnet\_link\_pf.c

src/libnet\_link\_snoop.c

Copyright: 1998-2004, Mike D. Schiffman <mike@infonexus.com>

1990-1996, 1998, The Regents of the University of California.

License: BSD-3

Files: src/libnet\_link\_snit.c

Copyright: 1998-2004, Mike D. Schiffman <mike@infonexus.com>

1990-1996, 1998, The Regents of the University of California.

1989, Micky Liu <micky@cunixc.cc.columbia.edu>, Columbia University

License: BSD-3

Files: src/libnet\_link\_win32.c

Copyright: 1998-2004, Mike D. Schiffman <mike@infonexus.com>

2001-2002, Don Bowman <don@sandvine.com>

2002, Roberto Larcher <roberto.larcher@libero.it>

License: BSD-2

Files: \*

Copyright: 1998-2004, Mike D. Schiffman <mike@infonexus.com>

License: BSD-2

Copyright (C) 2010 Wurldtech Security Technologies All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# 1.67 iputils 20240905-r0

## 1.67.1 Available under license :

arping: GPL-2.0-or-later  
clockdiff: BSD-3-Clause  
ping: BSD-3-Clause  
tracepath: GPL-2.0-or-later

Files containing license texts are available in Documentation directory.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This

General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a



notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such

an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues),

conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any

later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively

convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into

proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

## 1.68 iptraf-ng 1.2.2-r0

### 1.68.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their

rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate

copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.



In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program

except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is

implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.

EXCEPT WHEN

OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED

OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License,  
or  
(at your option) any later version.

This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

## 1.69 lua 5.4.7-r0

### 1.69.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
@Ci{$Id: manual.of $}
@C{[(-----)}
@manual{

@sect1{@title{Introduction}}
```

Lua is a powerful, efficient, lightweight, embeddable scripting language. It supports procedural programming, object-oriented programming, functional programming,

data-driven programming, and data description.

Lua combines simple procedural syntax with powerful data description constructs based on associative arrays and extensible semantics.

Lua is dynamically typed,  
runs by interpreting bytecode with a register-based  
virtual machine,  
and has automatic memory management with  
a generational garbage collection,  
making it ideal for configuration, scripting,  
and rapid prototyping.

Lua is implemented as a library, written in `clean C`,  
the common subset of `standard C` and `C++`.

The Lua distribution includes a host program called `lua`,  
which uses the Lua library to offer  
a complete,  
standalone Lua interpreter,  
for interactive or batch use.

Lua is intended to be used both as a powerful, lightweight,  
embeddable scripting language for any program that needs one,  
and as a powerful but lightweight and efficient stand-alone language.

As an extension language, Lua has no notion of a `main` program:  
it works `embedded` in a host client,  
called the `embedding program` or simply the `host`.  
(Frequently, this host is the stand-alone `lua` program.)

The host program can invoke functions to execute a piece of Lua code,  
can write and read Lua variables,  
and can register `C functions` to be called by Lua code.

Through the use of `C functions`, Lua can be augmented to cope with  
a wide range of different domains,  
thus creating customized programming languages sharing a syntactical framework.

Lua is free software,  
and is provided as usual with no guarantees,  
as stated in its license.

The implementation described in this manual is available  
at  
Lua's official web site, `www.lua.org`.

Like any other reference manual,  
this document is dry in places.

For a discussion of the decisions behind the design of Lua,  
see the technical papers available at Lua's web site.

For a detailed introduction to programming in Lua,  
see Roberto's book, `Programming in Lua`.

}

```
@C{-----}  
@sect1{basic| @title{Basic Concepts}
```

```
@simplesect{
```

This section describes the basic concepts of the language.

```
}
```

```
@sect2{TypesSec| @title{Values and Types}
```

Lua is a dynamically typed language.

This means that

variables do not have types; only values do.

There are no type definitions in the language.

All values carry their own type.

All values in Lua are first-class values.

This means that all values can be stored in variables,  
passed as arguments to other functions, and returned as results.

There are eight @x{basic types} in Lua:

@def{nil}, @def{boolean}, @def{number},

@def{string},

@def{function}, @def{userdata},

@def{thread}, and @def{table}.

The type @emph{nil} has one single value, @nil,

whose main property is to be different from any other value;

it often represents the absence of a useful value.

The type @emph{boolean} has two values, @false and @true.

Both @nil and @false make a condition false;

they are collectively called @def{false values}.

Any other value makes a condition true.

Despite its name,

@false is frequently used as an alternative to @nil,

with the key difference that @false behaves

like a regular value in a table,

while a @nil in a table represents an absent key.

The type @emph{number} represents both

integer numbers and real (floating-point) numbers,

using two @x{subtypes}: @def{integer} and @def{float}.

Standard Lua uses 64-bit integers and double-precision (64-bit) floats,

but you can also compile Lua so that it

uses 32-bit integers and/or single-precision (32-bit) floats.

The option with 32 bits for both integers and floats is particularly attractive for small machines and embedded systems. (See macro `@id{LUA_32BITS}` in file `@id{luaconf.h}`.)

Unless stated otherwise, any overflow when manipulating integer values `@def{wrap around}`, according to the usual rules of two-complement arithmetic. (In other words, the actual result is the unique representable integer that is equal modulo  $2^M$  to the mathematical result, where  $M$  is the number of bits of the integer type.)

Lua has explicit rules about when each subtype is used, but it also converts between them automatically as needed `@see{coercion}`. Therefore, the programmer may choose to mostly ignore the difference between integers and floats or to assume complete control over the representation of each number.

The type `@emph{string}` represents immutable sequences of bytes. `@index{eight-bit clean}`  
Lua is 8-bit clean:  
strings can contain any 8-bit value, including `@x{embedded zeros}` (`@Char{0}`).  
Lua is also encoding-agnostic;  
it makes no assumptions about the contents of a string.  
The length of any string in Lua must fit in a Lua integer.

Lua can call (and manipulate) functions written in Lua and functions written in C `@see{functioncall}`. Both are represented by the type `@emph{function}`.

The type `@emph{userdata}` is provided to allow arbitrary `@N{C data}` to be stored in Lua variables.  
A userdata value represents a block of raw memory.  
There are two kinds of userdata:  
`@emphx{full userdata}`, which is an object with a block of memory managed by Lua, and `@emphx{light userdata}`, which is simply a `@N{C pointer}` value.  
Userdata has no predefined operations in Lua, except assignment and identity test.  
By using `@emph{metatables}`, the programmer can define operations for full userdata values `@see{metatable}`.



Userdata values cannot be created or modified in Lua, only through the @N{C API}. This guarantees the integrity of data owned by the host program and @N{C libraries}.

The type @def{thread} represents independent threads of execution and it is used to implement coroutines @see{coroutine}. Lua threads are not related to operating-system threads. Lua supports coroutines on all systems, even those that do not support threads natively.

The type @emph{table} implements @x{associative arrays}, that is, @x{arrays} that can have as indices not only numbers, but any Lua value except @nil and @x{NaN}. (@emphx{Not a Number} is a special floating-point value used by the @x{IEEE 754} standard to represent undefined numerical results, such as @T{0/0}.) Tables can be @emph{heterogeneous}; that is, they can contain values of all types (except @nil). Any key associated to the value @nil is not considered part of the table. Conversely, any key that is not part of a table has an associated value @nil.

Tables are the sole data-structuring mechanism in Lua; they can be used to represent ordinary arrays, lists, symbol tables, sets, records, graphs, trees, etc. To represent @x{records}, Lua uses the field name as an index. The language supports this representation by providing @id{a.name} as syntactic sugar for @T{a["name"]}. There are several convenient ways to create tables in Lua @see{tableconstructor}.

Like indices, the values of table fields can be of any type. In particular, because functions are first-class values, table fields can contain functions. Thus tables can also carry @emph{methods} @see{func-def}.

The indexing of tables follows the definition of raw equality in the language. The expressions @T{a[i]} and @T{a[j]} denote the same table element if and only if @id{i} and @id{j} are raw equal (that is, equal without metamethods). In particular, floats with integral values

are equal to their respective integers  
(e.g., `@T{1.0 == 1}`).  
To avoid ambiguities,  
any float used as a key that is equal to an integer  
is converted to that integer.  
For instance, if you write `@T{a[2.0] = true}`,  
the actual key inserted into the table will be the integer `@T{2}`.

Tables, functions, threads, and (full) userdata values are `@emph{objects}`:  
variables  
do not actually `@emph{contain}` these values,  
only `@emph{references}` to them.  
Assignment, parameter passing, and function returns  
always manipulate references to such values;  
these operations do not imply any kind of copy.

The library function `@Lid{type}` returns a string describing the type  
of a given value `@seeF{type}`.

}

`@sect2{globalenv| @title{Environments and the Global Environment}`

As we will discuss further in `@refsec{variables}` and `@refsec{assignment}`,  
any reference to a free name  
(that is, a name not bound to any declaration) `@id{var}`  
is syntactically translated to `@T{_ENV.var}`.  
Moreover, every chunk is compiled in the scope of  
an external local variable named `@id{_ENV}` `@see{chunks}`,  
so `@id{_ENV}` itself is never a free name in a chunk.

Despite the existence of this external `@id{_ENV}` variable and  
the translation of free names,  
`@id{_ENV}` is a completely regular name.  
In particular,  
you can define new variables and parameters with that name.  
Each reference to a free  
name uses the `@id{_ENV}` that is  
visible at that point in the program,  
following the usual visibility rules of Lua `@see{visibility}`.

Any table used as the value of `@id{_ENV}` is called an `@def{environment}`.

Lua keeps a distinguished environment called the `@def{global environment}`.  
This value is kept at a special index in the C registry `@see{registry}`.  
In Lua, the global variable `@Lid{_G}` is initialized with this same value.  
(`@Lid{_G}` is never used internally,

so changing its value will affect only your own code.)

When Lua loads a chunk,  
the default value for its `@id{_ENV}` variable  
is the global environment `@seeF{load}`.  
Therefore, by default,  
free names in Lua code refer to entries in the global environment  
and, therefore, they are also called `@def{global variables}`.  
Moreover, all standard libraries are loaded in the global environment  
and some functions there operate on that environment.  
You can use `@Lid{load}` (or `@Lid{loadfile}`)  
to load a chunk with a different environment.  
(In C,  
you have to load the chunk and then change the value  
of its first upvalue; see `@See{lua_setupvalue}`.)

}

`@sect2{error| @title{Error Handling}`

Several operations in Lua can `@emph{raise}` an error.  
An error interrupts the normal flow of the program,  
which can continue by `@emph{catching}` the error.

Lua code can explicitly raise an error by calling the  
`@Lid{error}` function.  
(This function never returns.)

To catch errors in Lua,  
you can do a `@def{protected call}`,  
using `@Lid{pcall}` (or `@Lid{xpcall}`).  
The function `@Lid{pcall}` calls a given function in `@def{protected mode}`.  
Any error while running the function stops its execution,  
and control returns immediately to `@id{pcall}`,  
which returns a status code.

Because Lua is an embedded extension language,  
Lua code starts running by a call  
from `@N{C code}` in the host program.  
(When you use Lua standalone,  
the `@id{lua}` application is the host program.)  
Usually, this call is protected;  
so, when an otherwise unprotected error occurs during  
the compilation  
or execution of a Lua chunk,  
control returns to the host,  
which can take appropriate measures,  
such as printing an error message.

Whenever there is an error,  
an @def{error object}  
is propagated with information about the error.  
Lua itself only generates errors whose error object is a string,  
but programs can generate errors with  
any value as the error object.  
It is up to the Lua program or its host to handle such error objects.  
For historical reasons,  
an error object is often called an @def{error message},  
even though it does not have to be a string.

When you use @Lid{xpcall} (or @Lid{lua\_pcall}, in C)  
you can give a @def{message handler}  
to be called in case of errors.  
This function is called with the original error object  
and returns a new error object.  
It is called before the error unwinds the stack,  
so that it can gather more information about the error,  
for instance by inspecting the stack and creating a stack traceback.  
This message handler is still protected by the  
protected call;  
so, an error inside the message handler  
will call the message handler again.  
If this loop goes on for too long,  
Lua breaks it and returns an appropriate message.  
The message handler is called only for regular runtime errors.  
It is not called for memory-allocation errors  
nor for errors while running finalizers or other message handlers.

Lua also offers a system of @emph{warnings} @seeF{warn}.  
Unlike errors, warnings do not interfere  
in any way with program execution.  
They typically only generate a message to the user,  
although this behavior can be adapted from C @seeC{lua\_setwarnf}.

}

@sect2{metatable| @title{Metatables and Metamethods}}

Every value in Lua can have a @emph{metatable}.  
This @def{metatable} is an ordinary Lua table  
that defines the behavior of the original value  
under certain events.  
You can change several aspects of the behavior  
of a value by setting specific fields in its metatable.  
For instance, when a non-numeric value is the operand of an addition,

Lua

checks for a function in the field `@idx{__add}` of the value's metatable.

If it finds one,

Lua calls this function to perform the addition.

The key for each event in a metatable is a string

with the event name prefixed by two underscores;

the corresponding value is called a `@def{metavalue}`.

For most events, the metavalue must be a function,

which is then called a `@def{metamethod}`.

In the previous example, the key is the string `@St{__add}`

and the metamethod is the function that performs the addition.

Unless stated otherwise,

a metamethod can in fact be any `@x{callable value}`,

which is either a function or a value with a `@idx{__call}` metamethod.

You can query the metatable of any value

using the `@Lid{getmetatable}` function.

Lua queries metamethods in metatables using a raw access `@seeF{rawget}`.

You can replace the metatable of tables

using the `@Lid{setmetatable}` function.

You cannot change the metatable of other types from Lua code,

except by using the `@link{debuglib|debug library}`.

Tables

and full userdata have individual metatables,

although multiple tables and userdata can share their metatables.

Values of all other types share one single metatable per type;

that is, there is one single metatable for all numbers,

one for all strings, etc.

By default, a value has no metatable,

but the string library sets a metatable for the string type `@see{strlib}`.

A detailed list of operations controlled by metatables is given next.

Each event is identified by its corresponding key.

By convention, all metatable keys used by Lua are composed by

two underscores followed by lowercase Latin letters.

`@description{`

`@item{@idx{__add}|`

the addition (`@T{+}`) operation.

If any operand for an addition is not a number,

Lua will try to call a metamethod.

It starts by checking the first operand (even if it is a number);

if that operand does not define a metamethod for `@idx{__add}`,

then Lua will check the second operand.

If Lua can find a metamethod,  
it calls the metamethod with the two  
operands as arguments,  
and the result of the call  
(adjusted to one value)  
is the result of the operation.  
Otherwise, if no metamethod is found,  
Lua raises an error.  
}

@item{@idx{\_\_sub}|  
the subtraction (@T{-}) operation.  
Behavior similar to the addition operation.  
}

@item{@idx{\_\_mul}|  
the multiplication (@T{\*}) operation.  
Behavior similar to the addition operation.  
}

@item{@idx{\_\_div}|  
the division (@T{/}) operation.  
Behavior similar to the addition operation.  
}

@item{@idx{\_\_mod}|  
the modulo (@T{%}) operation.  
Behavior similar to the addition operation.  
}

@item{@idx{\_\_pow}|  
the exponentiation (@T{^}) operation.  
Behavior similar to the addition operation.  
}

@item{@idx{\_\_unm}|  
the negation (unary @T{-}) operation.  
Behavior similar to the addition operation.  
}

@item{@idx{\_\_idiv}|  
the floor division (@T{//}) operation.  
Behavior similar to the addition operation.  
}

@item{@idx{\_\_band}|  
the bitwise AND (@T{&}) operation.  
Behavior similar to the addition operation,

except that Lua will  
try a metamethod  
if any operand is neither an integer  
nor a float coercible to an integer @see{coercion}.  
}

@item{@idx{\_\_bor}|  
the bitwise OR (@T{||}) operation.  
Behavior similar to the bitwise AND operation.  
}

@item{@idx{\_\_bxor}|  
the bitwise exclusive OR (binary @T{~}) operation.  
Behavior similar to the bitwise AND operation.  
}

@item{@idx{\_\_bnot}|  
the bitwise NOT (unary @T{~}) operation.  
Behavior similar to the bitwise AND operation.  
}

@item{@idx{\_\_shl}|  
the bitwise left shift (@T{<<}) operation.  
Behavior similar to the bitwise AND operation.  
}

@item{@idx{\_\_shr}|  
the bitwise right shift (@T{>>}) operation.  
Behavior similar to the bitwise AND operation.  
}

@item{@idx{\_\_concat}|  
the concatenation (@T{..}) operation.  
Behavior similar to the addition operation,  
except that Lua will try a metamethod  
if any operand is neither a string nor a number  
(which is always coercible to a string).  
}

@item{@idx{\_\_len}|  
the length (@T{#}) operation.  
If the object is not a string,  
Lua will try its  
metamethod.  
If there is a metamethod,  
Lua calls it with the object as argument,  
and the result of the call  
(always adjusted to one value)

is the result of the operation.  
If there is no metamethod but the object is a table,  
then Lua uses the table length operation @see{len-op}.  
Otherwise, Lua raises an error.  
}

@item{@idx{\_\_eq}|  
the equal (@T{==}) operation.  
Behavior similar to the addition operation,  
except that Lua will try a metamethod only when the values  
being compared are either both tables or both full userdata  
and they are not primitively equal.  
The result of the call is always converted to a boolean.  
}

@item{@idx{\_\_lt}|  
the less than (@T{<}) operation.  
Behavior similar to the addition operation,  
except that Lua will try a metamethod only when the values  
being compared are neither both numbers nor both strings.  
Moreover, the result of the call is always converted to a boolean.  
}

@item{@idx{\_\_le}|  
the less equal (@T{<=}) operation.  
Behavior similar to the less than operation.  
}

@item{@idx{\_\_index}|  
The  
indexing access operation @T{table[key]}.  
This event happens when @id{table} is not a table or  
when @id{key} is not present in @id{table}.  
The metavalue is looked up in the metatable of @id{table}.

The metavalue for this event can be either a function, a table,  
or any value with an @idx{\_\_index} metavalue.  
If it is a function,  
it is called with @id{table} and @id{key} as arguments,  
and the result of the call  
(adjusted to one value)  
is the result of the operation.  
Otherwise,  
the final result is the result of indexing this metavalue with @id{key}.  
This indexing is regular, not raw,  
and therefore can trigger another @idx{\_\_index} metavalue.  
}



@item{@idx{\_\_newindex}|

The indexing assignment @T{table[key] = value}.

Like the index event,

this event happens when @id{table} is not a table or

when @id{key} is not present in @id{table}.

The metavalue is looked up in the metatable of @id{table}.

Like with indexing,

the metavalue for this event can be either a function,

a table,

or any value with an @idx{\_\_newindex} metavalue.

If it is a function,

it is called with @id{table}, @id{key}, and @id{value} as arguments.

Otherwise,

Lua repeats the indexing assignment over this metavalue

with the same key and value.

This assignment is regular, not raw,

and therefore can trigger another @idx{\_\_newindex} metavalue.

Whenever a @idx{\_\_newindex} metavalue is invoked,

Lua does not perform the primitive assignment.

If needed,

the metamethod itself can call @Lid{rawset}

to do the assignment.

}

@item{@idx{\_\_call}|

The call operation @T{func(args)}.

This event happens when Lua tries to call a non-function value

(that is, @id{func} is not a function).

The metamethod is looked up in @id{func}.

If present,

the metamethod is called with @id{func} as its first argument,

followed by the arguments of the original call (@id{args}).

All results of the call

are the results of the operation.

This is the only metamethod that allows multiple results.

}

}

In addition to

the previous list,

the interpreter also respects the following keys in metatables:

@idx{\_\_gc} @see{finalizers},

@idx{\_\_close} @see{to-be-closed},

@idx{\_\_mode} @see{weak-table},

and @idx{\_\_name}.

(The entry `@idx{__name}`,  
when it contains a string,  
may be used by `@Lid{tostring}` and in error messages.)

For the unary operators (negation, length, and bitwise NOT),  
the metamethod is computed and called with a dummy second operand,  
equal to the first one.

This extra operand is only to simplify Lua's internals  
(by making these operators behave like a binary operation)  
and may be removed in future versions.  
For most uses this extra operand is irrelevant.

Because metatables are regular tables,  
they can contain arbitrary fields,  
not only the event names defined above.  
Some functions in the standard library  
(e.g., `@Lid{tostring}`)  
use other fields in metatables for their own purposes.

It is a good practice to add all needed metamethods to a table  
before setting it as a metatable of some object.

In  
particular, the `@idx{__gc}` metamethod works only when this order  
is followed `@see{finalizers}`.

It is also a good practice to set the metatable of an object  
right after its creation.

}

`@sect2{GC| @title{Garbage Collection}`

`@simplesect{`

Lua performs automatic memory management.  
This means that  
you do not have to worry about allocating memory for new objects  
or freeing it when the objects are no longer needed.  
Lua manages memory automatically by running  
a `@def{garbage collector}` to collect all `@emph{dead}` objects.  
All memory used by Lua is subject to automatic management:  
strings, tables, userdata, functions, threads, internal structures, etc.

An object is considered `@def{dead}`  
as soon as the collector can be sure the object  
will not be accessed again in the normal execution of the program.  
(`@Q{Normal execution}` here excludes finalizers,  
which can resurrect dead objects `@see{finalizers}`,  
and excludes also operations using the debug library.)

Note that the time when the collector can be sure that an object is dead may not coincide with the programmer's expectations. The only guarantees are that Lua will not collect an object that may still be accessed in the normal execution of the program, and it will eventually collect an object that is inaccessible from Lua. (Here, `@emph{inaccessible from Lua}` means that neither a variable nor another live object refer to the object.) Because Lua has no knowledge about `@N{C code}`, it never collects objects accessible through the registry `@see{registry}`, which includes the global environment `@see{globalenv}`.

The garbage collector (GC) in Lua can work in two modes: incremental and generational.

The default GC mode with the default parameters are adequate for most uses. However, programs that waste a large proportion of their time allocating and freeing memory can benefit from other settings. Keep in mind that the GC behavior is non-portable both across platforms and across different Lua releases; therefore, optimal settings are also non-portable.

You can change the GC mode and parameters by calling `@Lid{lua_gc} @N{in C}` or `@Lid{collectgarbage}` in Lua. You can also use these functions to control the collector directly (e.g., to stop and restart it).

}

`@sect3{incmode| @title{Incremental Garbage Collection}`

In incremental mode, each GC cycle performs a mark-and-sweep collection in small steps interleaved with the program's execution. In this mode, the collector uses three numbers to control its garbage-collection cycles: the `@def{garbage-collector pause}`, the `@def{garbage-collector step multiplier}`, and the `@def{garbage-collector step size}`.

The garbage-collector pause controls how long the collector waits before starting a new cycle.

The collector starts a new cycle when the use of memory hits `@M{n%}` of the use after the previous collection. Larger values make the collector less aggressive. Values equal to or less than 100 mean the collector will not wait to start a new cycle. A value of 200 means that the collector waits for the total memory in use to double before starting a new cycle. The default value is 200; the maximum value is 1000.

The garbage-collector step multiplier controls the speed of the collector relative to memory allocation, that is, how many elements it marks or sweeps for each kilobyte of memory allocated. Larger values make the collector more aggressive but also increase the size of each incremental step. You should not use values less than 100, because they make the collector too slow and can result in the collector never finishing a cycle. The default value is 100; the maximum value is 1000.

The garbage-collector step size controls the size of each incremental step, specifically how many bytes the interpreter allocates before performing a step. This parameter is logarithmic: A value of `@M{n}` means the interpreter will allocate `@M{2@sp{n}}` bytes between steps and perform equivalent work during the step. A large value (e.g., 60) makes the collector a stop-the-world (non-incremental) collector. The default value is 13, which means steps of approximately `@N{8 Kbytes}`.

}

@sect3{genmode| @title{Generational Garbage Collection}

In generational mode, the collector does frequent `@emph{minor}` collections, which traverses only objects recently created. If after a minor collection the use of memory is still above a limit, the collector does a stop-the-world `@emph{major}` collection, which traverses all objects. The generational mode uses two parameters: the `@def{minor multiplier}` and the `@def{the major multiplier}`.

The minor multiplier controls the frequency of minor collections.  
 For a minor multiplier `@M{x}`,  
 a new minor collection will be done when memory  
 grows `@M{x%}` larger than the memory in use after the previous major  
 collection.  
 For instance, for a multiplier of 20,  
 the collector will do a minor collection when the use of memory  
 gets 20% larger than the use after the previous major collection.  
 The default value is 20; the maximum value is 200.

The  
 major multiplier controls the frequency of major collections.  
 For a major multiplier `@M{x}`,  
 a new major collection will be done when memory  
 grows `@M{x%}` larger than the memory in use after the previous major  
 collection.  
 For instance, for a multiplier of 100,  
 the collector will do a major collection when the use of memory  
 gets larger than twice the use after the previous collection.  
 The default value is 100; the maximum value is 1000.

}

`@sect3{finalizers| @title{Garbage-Collection Metamethods}`

You can set garbage-collector metamethods for tables  
 and, using the `@N{C API}`,  
 for full userdata `@see{metatable}`.  
 These metamethods, called `@def{finalizers}`,  
 are called when the garbage collector detects that the  
 corresponding table or userdata is dead.  
 Finalizers allow you to coordinate Lua's garbage collection  
 with external resource management such as closing files,  
 network or database connections,  
 or freeing your own memory.

For an object (table or userdata) to be finalized when collected,  
 you  
 must `@emph{mark}` it for finalization.  
`@index{mark (for finalization)}`  
 You mark an object for finalization when you set its metatable  
 and the metatable has a `@idx{__gc}` metamethod.  
 Note that if you set a metatable without a `@idx{__gc}` field  
 and later create that field in the metatable,  
 the object will not be marked for finalization.

When a marked object becomes dead,

it is not collected immediately by the garbage collector.  
Instead, Lua puts it in a list.  
After the collection,  
Lua goes through that list.  
For each object in the list,  
it checks the object's `@idx{__gc}` metamethod:  
If it is present,  
Lua calls it with the object as its single argument.

At the end of each garbage-collection cycle,  
the finalizers are called in  
the reverse order that the objects were marked for finalization,  
among those collected in that cycle;  
that is, the first finalizer to be called is the one associated  
with the object marked last in the program.  
The execution of each finalizer may occur at any  
point during  
the execution of the regular code.

Because the object being collected must still be used by the finalizer,  
that object (and other objects accessible only through it)  
must be *@emph{resurrected}* by `Lua.@index{resurrection}`  
Usually, this resurrection is transient,  
and the object memory is freed in the next garbage-collection cycle.  
However, if the finalizer stores the object in some global place  
(e.g., a global variable),  
then the resurrection is permanent.  
Moreover, if the finalizer marks a finalizing object for finalization again,  
its finalizer will be called again in the next cycle where the  
object is dead.  
In any case,  
the object memory is freed only in a GC cycle where  
the object is dead and not marked for finalization.

When you close a state `@seeF{lua_close}`,  
Lua calls the finalizers of all objects marked for finalization,  
following the reverse order that they were marked.  
If any finalizer marks objects for collection during that phase,  
these marks have no effect.

#### Finalizers

cannot yield nor run the garbage collector.  
Because they can run in unpredictable times,  
it is good practice to restrict each finalizer  
to the minimum necessary to properly release  
its associated resource.

Any error while running a finalizer generates a warning;

the error is not propagated.

}

@sect3{weak-table| @title{Weak Tables}

A @def{weak table} is a table whose elements are  
@def{weak references}.

A weak reference is ignored by the garbage collector.

In other words,

if the only references to an object are weak references,  
then the garbage collector will collect that object.

A weak table can have weak keys, weak values, or both.

A table with weak values allows the collection of its values,  
but prevents the collection of its keys.

A table with both weak keys and weak values allows the collection of  
both keys and values.

In any case, if either the key or the value is collected,  
the whole pair is removed from the table.

The weakness of a table is controlled by the  
@idx{\_\_mode}  
field of its metatable.

This metavalue, if present, must be one of the following strings:

@St{k}, for a table with weak keys;

@St{v}, for a table with weak values;

or @St{kv}, for a table with both weak keys and values.

A table with weak keys and strong values  
is also called an @def{ephemeron table}.

In an ephemeron table,

a value is considered reachable only if its key is reachable.

In particular,

if the only reference to a key comes through its value,  
the pair is removed.

Any change in the weakness of a table may take effect only  
at the next collect cycle.

In particular, if you change the weakness to a stronger mode,  
Lua may still collect some items from that table  
before the change takes effect.

Only objects that have an explicit construction  
are removed from weak tables.

Values, such as numbers and @x{light @N{C functions}},  
are not subject to garbage collection,  
and therefore are not removed from weak tables  
(unless their associated values are collected).

Although strings  
are subject to garbage collection,  
they do not have an explicit construction and  
their equality is by value;  
they behave more like values than like objects.  
Therefore, they are not removed from weak tables.

Resurrected objects  
(that is, objects being finalized  
and objects accessible only through objects being finalized)  
have a special behavior in weak tables.  
They are removed from weak values before running their finalizers,  
but are removed from weak keys only in the next collection  
after running their finalizers, when such objects are actually freed.  
This behavior allows the finalizer to access properties  
associated with the object through weak tables.

If a weak table is among the resurrected objects in a collection cycle,  
it may not be properly cleared until the next cycle.

}

}

@sect2{coroutine| @title{Coroutines}

Lua supports coroutines,  
also called @emphx{collaborative multithreading}.  
A coroutine in Lua represents an independent thread of execution.  
Unlike threads in multithread  
systems, however,  
a coroutine only suspends its execution by explicitly calling  
a yield function.

You create a coroutine by calling @Lid{coroutine.create}.  
Its sole argument is a function  
that is the main function of the coroutine.  
The @id{create} function only creates a new coroutine and  
returns a handle to it (an object of type @emph{thread});  
it does not start the coroutine.

You execute a coroutine by calling @Lid{coroutine.resume}.  
When you first call @Lid{coroutine.resume},  
passing as its first argument  
a thread returned by @Lid{coroutine.create},  
the coroutine starts its execution by  
calling its main function.  
Extra arguments passed to @Lid{coroutine.resume} are passed



as arguments to that function.  
After the coroutine starts running,  
it runs until it terminates or `@emph{yields}`.

A coroutine can terminate its execution in two ways:  
normally, when its main function returns  
(explicitly or implicitly, after the last instruction);  
and abnormally, if there is an unprotected  
error.

In case of normal termination,  
`@Lid{coroutine.resume}` returns `@true`,  
plus any values returned by the coroutine main function.  
In case of errors, `@Lid{coroutine.resume}` returns `@false`  
plus the error object.  
In this case, the coroutine does not unwind its stack,  
so that it is possible to inspect it after the error  
with the debug API.

A coroutine yields by calling `@Lid{coroutine.yield}`.  
When a coroutine yields,  
the corresponding `@Lid{coroutine.resume}` returns immediately,  
even if the yield happens inside nested function calls  
(that is, not in the main function,  
but in a function directly or indirectly called by the main function).  
In the case of a yield, `@Lid{coroutine.resume}` also returns `@true`,  
plus any values passed to `@Lid{coroutine.yield}`.  
The next time you resume the same coroutine,  
it continues its execution from the point where it yielded,  
with the call to `@Lid{coroutine.yield}` returning any extra  
arguments passed to `@Lid{coroutine.resume}`.

Like `@Lid{coroutine.create}`,  
the  
`@Lid{coroutine.wrap}` function also creates a coroutine,  
but instead of returning the coroutine itself,  
it returns a function that, when called, resumes the coroutine.  
Any arguments passed to this function  
go as extra arguments to `@Lid{coroutine.resume}`.  
`@Lid{coroutine.wrap}` returns all the values returned by `@Lid{coroutine.resume}`,  
except the first one (the boolean error code).  
Unlike `@Lid{coroutine.resume}`,  
the function created by `@Lid{coroutine.wrap}`  
propagates any error to the caller.  
In this case,  
the function also closes the coroutine `@seeF{coroutine.close}`.

As an example of how coroutines work,  
consider the following code:

```

@verbatim{
function foo (a)
  print("foo", a)
  return coroutine.yield(2*a)
end

co = coroutine.create(function (a,b)
  print("co-body", a, b)
  local r = foo(a+1)
  print("co-body", r)
  local r, s = coroutine.yield(a+b, a-b)
  print("co-body", r, s)
  return b, "end"
end)

print("main", coroutine.resume(co, 1, 10))
print("main", coroutine.resume(co,
  "r"))
print("main", coroutine.resume(co, "x", "y"))
print("main", coroutine.resume(co, "x", "y"))
}

```

When you run it, it produces the following output:

```

@verbatim{
co-body 1    10
foo    2
main   true   4
co-body r
main   true  11  -9
co-body x    y
main   true  10   end
main   false cannot resume dead coroutine
}

```

You can also create and manipulate coroutines through the C API:  
 see functions @Lid{lua\_newthread}, @Lid{lua\_resume},  
 and @Lid{lua\_yield}.

```

}

```

```

}

```

```

@C{-----}
@sect1{language| @title{The Language}

```

```

@simplesect{

```

This section describes the lexis, the syntax, and the semantics of Lua.

In other words,  
this section describes  
which tokens are valid,  
how they can be combined,  
and what their combinations mean.

Language constructs will be explained using the usual extended BNF notation,  
in which

@N{@bnfrep{@rep{a}} means 0} or more @rep{a}'s, and  
@N{@bnfopt{@rep{a}} means} an  
optional @rep{a}.

Non-terminals are shown like @bnfNter{non-terminal},

keywords are shown like @rw{kword},

and other terminal symbols are shown like @bnfter{=}.

The complete syntax of Lua can be found in @refsec{BNF}  
at the end of this manual.

}

@sect2{lexical| @title{Lexical Conventions}

Lua is a @x{free-form} language.

It ignores spaces and comments between lexical elements (@x{tokens}),  
except as delimiters between two tokens.

In source code,

Lua recognizes as spaces the standard ASCII whitespace  
characters space, form feed, newline,  
carriage return, horizontal tab, and vertical tab.

@def{Names}

(also called @def{identifiers})

in Lua can be any string of Latin letters,

Arabic-Indic digits, and underscores,

not beginning with a digit and

not being a reserved word.

Identifiers are used to name variables, table fields, and labels.

The following @def{keywords} are reserved  
and cannot be used as names:

@index{reserved words}

@verbatim{

and    break    do       else    elseif  
      end

false   for    function goto   if    in

local   nil    not    or    repeat   return

then    true    until   while

}

Lua is a case-sensitive language:

@id{and} is a reserved word, but @id{And} and @id{AND} are two different, valid names.

As a convention,

programs should avoid creating

names that start with an underscore followed by

one or more uppercase letters (such as @Lid{\_VERSION}).

The following strings denote other @x{tokens}:

```
@verbatim{
+   -   *   /   %   ^   #
&   ~   |   <<  >>  //
==  ~=  <=  >=  <   >   =
(   )   {   }   [   ]   ::
;   :   ,   .   ..  ...
}
```

A @def{short literal string}

can be delimited by matching single or double quotes,

and can contain the following C-like escape sequences:

@Char{a} (bell),

@Char{b} (backspace),

@Char{f} (form feed),

@Char{n} (newline),

@Char{r} (carriage return),

@Char{t} (horizontal tab),

@Char{v} (vertical tab),

@Char{\\} (backslash),

@Char{\"}

(quotation mark [double quote]),

and @Char{'} (apostrophe [single quote]).

A backslash followed by a line break

results in a newline in the string.

The escape sequence @Char{z} skips the following span

of whitespace characters,

including line breaks;

it is particularly useful to break and indent a long literal string

into multiple lines without adding the newlines and spaces

into the string contents.

A short literal string cannot contain unescaped line breaks

nor escapes not forming a valid escape sequence.

We can specify any byte in a short literal string,

including @x{embedded zeros},

by its numeric value.

This can be done

with the escape sequence @T{x@rep{XX}},

where @rep{XX} is a sequence of exactly two hexadecimal digits,

or with the escape sequence `@T{\@rep{ddd}}`,  
where `@rep{ddd}` is a sequence of up to three decimal digits.  
(Note that if a decimal escape sequence is to be followed by a digit,  
it must be expressed using exactly three digits.)

The `@x{UTF-8}` encoding  
of a `@x{Unicode}` character  
can be inserted in a literal string with  
the escape sequence `@T{\u{@rep{XXX}}}`  
(with mandatory enclosing braces),  
where `@rep{XXX}` is a sequence of one or more hexadecimal digits  
representing the character code point.  
This code point can be any value less than `@M{2@sp{31}}`.  
(Lua uses the original UTF-8 specification here,  
which is not restricted to valid Unicode code points.)

Literal strings can also be defined using a long format  
enclosed by `@def{long brackets}`.  
We define an `@def{opening long bracket of level @rep{n}}` as an opening  
square bracket followed by `@rep{n}` equal signs followed by another  
opening square bracket.  
So, an opening long bracket of `@N{level 0}` is written as `@T{[[, @C{]]}`  
an opening long bracket of `@N{level 1}` is written as `@T{[=[, @C{]]}`  
and so on.  
A `@emph{closing long bracket}` is defined similarly;  
for instance,  
a closing long bracket of `@N{level 4}` is written as `@C{[[ @T{]===}]}`.  
A `@def{long literal}` starts with an opening long bracket  
of any level and  
ends at the first closing long bracket of the same level.  
It can contain any text except a closing bracket of the same level.  
Literals in this bracketed form can run for several lines,  
do not interpret any escape sequences,  
and ignore long brackets of any other level.  
Any kind of end-of-line sequence  
(carriage return, newline, carriage return followed by newline,  
or newline followed by carriage return)  
is converted to a simple newline.  
When the opening long bracket is immediately followed by a newline,  
the newline is not included in the string.

As an example, in a system using ASCII  
(in which `@Char{a}` is coded `@N{as 97}`,  
newline is coded `@N{as 10}`, and `@Char{1}` is coded `@N{as 49}`),  
the five literal strings below denote the same string:  
`@verbatim{`  
`a = 'alo\n123''`  
`a = "alo\n123\"`

```

a = "\97lo\10\04923"
a = [[alo
123"]]
a = [=[
alo
123"]]=]
}

```

Any byte in a literal string not explicitly affected by the previous rules represents itself. However, Lua opens files for parsing in text mode, and the system's file functions may have problems with some control characters. So, it is safer to represent binary data as a quoted literal with explicit escape sequences for the non-text characters.

A `@def{numeric constant}` (or `@def{numeral}`) can be written with an optional fractional part and an optional decimal exponent, marked by a letter `@Char{e}` or `@Char{E}`. Lua also accepts `@x{hexadecimal constants}`, which start with `@T{0x}` or `@T{0X}`. Hexadecimal constants also accept an optional fractional part plus an optional binary exponent, marked by a letter `@Char{p}` or `@Char{P}` and written in decimal. (For instance, `@T{0x1.fp10}` denotes 1984, which is `@M{0x1f / 16}` multiplied by `@M{2@sp{10}}.`)

A numeric constant with a radix point or an exponent denotes a float; otherwise, if its value fits in an integer or it is a hexadecimal constant, it denotes an integer; otherwise (that is, a decimal integer numeral that overflows), it denotes a float. Hexadecimal numerals with neither a radix point nor an exponent always denote an integer value; if the value overflows, it `@emph{wraps around}` to fit into a valid integer.

Examples of valid integer constants are

```

@verbatim{
3 345 0xff 0xBEBADA
}

```

Examples of valid float constants are

```
@verbatim{
3.0  3.1416  314.16e-2  0.31416E1  34e1
0x0.1E  0xA23p-4  0X1.921FB54442D18P+1
}
```

A `@def{comment}` starts with a double hyphen (`@T{--}`) anywhere outside a string.

If the text immediately after `@T{--}` is not an opening long bracket, the comment is a `@def{short comment}`, which runs until the end of the line.

Otherwise, it is a `@def{long comment}`, which runs until the corresponding closing long bracket.

```
}
```

```
@sect2{variables| @title{Variables}}
```

Variables are places that store values.

There are three kinds of variables in Lua:

global variables, local variables, and table fields.

A single name can denote a global variable or a local variable (or a function's formal parameter,

which is a particular kind of

local variable):

```
@Produc{
@producname{var}@producbody{@bnfNter{Name}}
}
```

`@bnfNter{Name}` denotes identifiers `@see{lexical}`.

Any variable name is assumed to be global unless explicitly declared as a local `@see{localvar}`.

`@x{Local variables}` are `@emph{lexically scoped}`:

local variables can be freely accessed by functions

defined inside their scope `@see{visibility}`.

Before the first assignment to a variable, its value is `@nil`.

Square brackets are used to index a table:

```
@Produc{
@producname{var}@producbody{prefixexp @bnfter{[] exp @bnfter{[]}}
}
```

The meaning of accesses to table fields can be changed via metatables

`@see{metatable}`.

The syntax `@id{var.Name}` is just syntactic sugar for

```
@T{var["Name"]};
```

```
@Produc{
```

```
@producname{var}@producbody{prefixexp @bnfter{.} @bnfNter{Name}}
}
```

An access to a global variable `@id{x}`  
is equivalent to `@id{_ENV.x}`.

Due to the way that chunks are compiled,  
the variable `@id{_ENV}` itself is never global `@see{globalenv}`.

```
}
```

```
@sect2{stats| @title{Statements}}
```

```
@simplesect{
```

Lua

supports an almost conventional set of `@x{statements}`,  
similar to those in other conventional languages.

This set includes

blocks, assignments, control structures, function calls,  
and variable declarations.

```
}
```

```
@sect3{@title{Blocks}}
```

A `@x{block}` is a list of statements,  
which are executed sequentially:

```
@Produc{
@producname{block}@producbody{@bnfrep{stat}}
}
```

Lua has `@def{empty statements}`  
that allow you to separate statements with semicolons,  
start a block with a semicolon  
or write two semicolons in sequence:

```
@Produc{
@producname{stat}@producbody{@bnfter{;}}
}
```

Both function calls and assignments  
can start with an open parenthesis.

This possibility leads to an ambiguity in Lua's grammar.

Consider the following fragment:

```
@verbatim{
a = b + c
(print or io.write)('done')
}
```

The grammar could see this fragment in two ways:

```
@verbatim{
```



```
a = b + c(print or io.write)('done')
```

```
a = b + c; (print or io.write)('done')
}
```

The current parser always sees such constructions in the first way,

interpreting

the open parenthesis

as the start of the arguments to a call.

To avoid this ambiguity,

it is a good practice to always precede with a semicolon statements that start with a parenthesis:

```
@verbatim{
;(print or io.write)('done')
}
```

A block can be explicitly delimited to produce a single statement:

```
@Produc{
@producname{stat}@producbody{@Rw{do} block @Rw{end}}
}
```

Explicit blocks are useful

to control the scope of variable declarations.

Explicit blocks are also sometimes used to add a `@Rw{return}` statement in the middle of another block `@see{control}`.

```
}
```

```
@sect3{chunks| @title{Chunks}
```

The unit of compilation of Lua is called a `@def{chunk}`.

Syntactically,

a chunk is simply a block:

```
@Produc{
@producname{chunk}@producbody{block}
}
```

Lua handles a chunk as the body of an anonymous function with a variable number of arguments

`@see{func-def}`.

As such, chunks can define local variables, receive arguments, and return values.

Moreover, such anonymous function is compiled as in the scope of

an external local variable called `@id{_ENV}` `@see{globalenv}`.

The resulting function always has `@id{_ENV}` as its only external variable, even if it does not use that variable.

A chunk can be stored in a file or in a string inside the host program.

To execute a chunk,

Lua first *loads* it,

precompiling the chunk's code into instructions for a virtual machine,

and then Lua executes the compiled code

with an interpreter for the virtual machine.

Chunks can also be precompiled into binary form;

see the program `luac` and the function `string.dump` for details.

Programs in source and compiled forms are interchangeable;

Lua automatically detects the file type and acts accordingly *load*.

}

`@sect3{assignment| @title{Assignment}}`

Lua allows *multiple assignments*.

Therefore, the syntax for assignment

defines a list of variables on the left side

and a list of expressions on the right side.

The elements in both lists are separated by commas:

`@Produc{`

`@producname{stat}@producbody{varlist`

`@bnfter{=} explist`

`@producname{varlist}@producbody{var @bnfrep{@bnfter{,} var}}`

`@producname{explist}@producbody{exp @bnfrep{@bnfter{,} exp}}`

`}`

Expressions are discussed in *See{expressions}*.

Before the assignment,

the list of values is *adjusted* to the length of

the list of variables *multires*.

If a variable is both assigned and read

inside a multiple assignment,

Lua ensures that all reads get the value of the variable

before the assignment.

Thus the code

`@verbatim{`

`i = 3`

`i, a[i] = i+1, 20`

`}`

sets `a[3]` to 20, without affecting `a[4]`

because the `i` in `a[i]` is evaluated (to 3)

before it is assigned 4.

Similarly, the line

`@verbatim{`

`x, y = y, x`

```

}
exchanges the values of @id{x} and @id{y},
and
@verbatim{
x, y, z = y, z, x
}
cyclically permutes the values of @id{x}, @id{y}, and @id{z}.

```

Note that this guarantee covers only accesses syntactically inside the assignment statement. If a function or a metamethod called during the assignment changes the value of a variable, Lua gives no guarantees about the order of that access.

An assignment to a global name `@T{x = val}` is equivalent to the assignment `@T{_ENV.x = val} @see{globalenv}`.

The meaning of assignments to table fields and global variables (which are actually table fields, too) can be changed via metatables `@see{metatable}`.

```

}

```

```

@sect3{control| @title{Control Structures}
The control structures
@Rw{if}, @Rw{while}, and @Rw{repeat} have the usual meaning and
familiar syntax:
@index{while-do statement}
@index{repeat-until statement}
@index{if-then-else statement}
@Produc{
@producname{stat}@produciblebody{@Rw{while} exp @Rw{do} block @Rw{end}}
@producname{stat}@produciblebody{@Rw{repeat} block @Rw{until} exp}
@producname{stat}@produciblebody{@Rw{if} exp @Rw{then} block
@bnfrep{@Rw{elseif} exp @Rw{then} block}
@bnfoft{@Rw{else} block} @Rw{end}}
}

```

Lua also has a `@Rw{for}` statement, in two flavors `@see{for}`.

The `@x{condition expression}` of a control structure can return any value.

Both

`@false` and `@nil` test false.

All values different from `@nil` and `@false` test true.

In particular, the number 0 and the empty string also test true.

In the `@Rw{repeat}@En@Rw{until}` loop,  
the inner block does not end at the `@Rw{until}` keyword,  
but only after the condition.  
So, the condition can refer to local variables  
declared inside the loop block.

The `@Rw{goto}` statement transfers the program control to a label.

For syntactical reasons,

labels in Lua are considered statements too:

```
@index{goto statement}
```

```
@index{label}
```

```
@Produc{
```

```
@producname{stat}@producbody{@Rw{goto} Name}
```

```
@producname{stat}@producbody{label}
```

```
@producname{label}@producbody{@bnfter{::} Name @bnfter{::}}
```

```
}
```

A label is visible in the entire block where it is defined,  
except inside nested functions.

A goto can jump to any visible label as long as it does not  
enter into the scope of a local variable.

A label should not be declared

where a label with the same name is visible,

even if this other label has been declared in an

enclosing block.

The `@Rw{break}` statement terminates the execution of a

`@Rw{while}`, `@Rw{repeat}`, or `@Rw{for}` loop,

skipping to the next statement after the loop:

```
@index{break statement}
```

```
@Produc{
```

```
@producname{stat}@producbody{@Rw{break}}
```

```
}
```

A `@Rw{break}` ends the innermost enclosing loop.

The `@Rw{return}` statement is used to return values

from a function or a chunk

(which is handled as an anonymous function).

```
@index{return statement}
```

Functions can return more than one value,

so the syntax for the `@Rw{return}` statement is

```
@Produc{
```

```
@producname{stat}@producbody{@Rw{return} @bnfopt{explist} @bnfopt{@bnfter{;}}}
```

```
}
```

The `@Rw{return}` statement can only be written

as the last statement of a block.

If it is necessary to `@Rw{return}` in the middle of a block,

then an explicit inner block can be used,  
as in the idiom `@T{do return end}`,  
because now `@Rw{return}` is the last statement in its (inner) block.

}

@sect3{for| @title{For Statement}

@index{for statement}

The `@Rw{for}` statement has two forms:

one

numerical and one generic.

@sect4{@title{The numerical @Rw{for} loop}

The numerical `@Rw{for}` loop repeats a block of code while a  
control variable goes through an arithmetic progression.

It has the following syntax:

```
@Produc{
@producname{stat}@producbody{@Rw{for} @bnfNter{Name} @bnfter{=}
exp @bnfter{,} exp @bnfopt{@bnfter{,} exp} @Rw{do} block @Rw{end}}
}
```

The given identifier (`@bnfNter{Name}`) defines the control variable,  
which is a new variable local to the loop body (`@emph{block}`).

The loop starts by evaluating once the three control expressions.

Their values are called respectively

the `@emph{initial value}`, the `@emph{limit}`, and the `@emph{step}`.

If the step is absent, it defaults `@N{to 1}`.

If both the initial value and the step are integers,

the loop is done with integers;

note that the limit may not be an integer.

Otherwise, the three values are converted to

floats and the loop is done with floats.

Beware of floating-point accuracy in this case.

After that initialization,

the

loop body is repeated with the value of the control variable

going through an arithmetic progression,

starting at the initial value,

with a common difference given by the step.

A negative step makes a decreasing sequence;

a step equal to zero raises an error.

The loop continues while the value is less than

or equal to the limit

(greater than or equal to for a negative step).

If the initial value is already greater than the limit  
(or less than, if the step is negative),  
the body is not executed.

For integer loops,  
the control variable never wraps around;  
instead, the loop ends in case of an overflow.

You should not change the value of the control variable  
during the loop.

If you need its value after the loop,  
assign it to another variable before exiting the loop.

```
}
```

```
@sect4{@title{The generic @Rw{for} loop}
```

The generic @Rw{for} statement works over functions,  
called @def{iterators}.

On each iteration, the iterator function is called to produce a new value,  
stopping when this new  
value is @nil.

The generic @Rw{for} loop has the following syntax:

```
@Produc{  
  @producname{stat}@produciblebody{@Rw{for} namelist @Rw{in} explist  
    @Rw{do} block @Rw{end}}  
  @producname{namelist}@produciblebody{@bnfNter{Name} @bnfrep{@bnfter{,} @bnfNter{Name}}}  
}
```

A @Rw{for} statement like

```
@verbatim{  
  for @rep{var_1}, @Cdots, @rep{var_n} in @rep{explist} do @rep{body} end  
}
```

works as follows.

The names @rep{var\_i} declare loop variables local to the loop body.

The first of these variables is the @emph{control variable}.

The loop starts by evaluating @rep{explist}  
to produce four values:

an @emph{iterator function},  
a @emph{state},  
an initial value for the control variable,  
and a @emph{closing value}.

Then, at each iteration,

Lua calls the iterator function with two arguments:  
the state and the control variable.

The results from this call are then assigned to the loop variables, following the rules of multiple assignments @see{assignment}. If the control variable becomes @nil, the loop terminates. Otherwise, the body is executed and the loop goes to the next iteration.

The closing value behaves like a to-be-closed variable @see{to-be-closed}, which can be used to release resources when the loop ends. Otherwise, it does not interfere with the loop.

You should not change the value of the control variable during the loop.

```
}
}
```

@sect3{funcstat| @title{Function Calls as Statements}}

To allow possible side-effects, function calls can be executed as statements:

```
@Produc{
@producname{stat}@producbody{functioncall}
}
```

In this case, all returned values are thrown away. Function calls are explained in @See{functioncall}.

```
}
```

@sect3{localvar| @title{Local Declarations}}

@x{Local variables} can be declared anywhere inside a block.

The declaration can include an initialization:

```
@Produc{
@producname{stat}@producbody{@Rw{local} attnamelist @bnfopt{@bnfter{=} explist}}
@producname{attnamelist}@producbody{
@bnfNter{Name} attrib @bnfrep{@bnfter{,} @bnfNter{Name} attrib}}
}
```

If

present, an initial assignment has the same semantics of a multiple assignment @see{assignment}. Otherwise, all variables are initialized with @nil.

Each variable name may be postfixed by an attribute (a name between angle brackets):

```
@Produc{
@producname{attrib}@producbody{@bnfopt{@bnfter{<} @bnfNter{Name} @bnfter{>}}}
```

}

There are two possible attributes:

@id{const}, which declares a @x{constant variable},

that is, a variable that cannot be assigned to  
after its initialization;

and @id{close}, which declares a to-be-closed variable @see{to-be-closed}.

A list of variables can contain at most one to-be-closed variable.

A chunk is also a block @see{chunks},

and so local variables can be declared in a chunk outside any explicit block.

The visibility rules for local variables are explained in @See{visibility}.

}

@sect3{to-be-closed| @title{To-be-closed Variables}}

A to-be-closed variable behaves like a constant local variable,  
except that its value is @emph{closed} whenever the variable  
goes

out of scope, including normal block termination,  
exiting its block by @Rw{break}/@Rw{goto}/@Rw{return},  
or exiting by an error.

Here, to @emph{close} a value means

to call its @idx{\_\_close} metamethod.

When calling the metamethod,

the value itself is passed as the first argument

and the error object that caused the exit (if any)

is passed as a second argument;

if there was no error, the second argument is @nil.

The value assigned to a to-be-closed variable

must have a @idx{\_\_close} metamethod

or be a false value.

(@nil and @false are ignored as to-be-closed values.)

If several to-be-closed variables go out of scope at the same event,  
they are closed in the reverse order that they were declared.

If there is any error while running a closing method,  
that error is handled like an error in the regular code  
where the variable was defined.

After an error,

the other pending closing methods will still be called.

If a coroutine yields and is never resumed again,  
some variables may



never go out of scope,  
and therefore they will never be closed.  
(These variables are the ones created inside the coroutine  
and in scope at the point where the coroutine yielded.)  
Similarly, if a coroutine ends with an error,  
it does not unwind its stack,  
so it does not close any variable.  
In both cases,  
you can either use finalizers  
or call `@Lid{coroutine.close}` to close the variables.  
However, if the coroutine was created  
through `@Lid{coroutine.wrap}`,  
then its corresponding function will close the coroutine  
in case of errors.

```
}
```

```
}
```

```
@sect2{expressions| @title{Expressions}}
```

```
@simplesect{
```

The basic expressions in Lua are the following:

```
@Produc{
@producname{exp}@producbody{prefixexp}
@producname{exp}@producbody{@Rw{nil} @Or @Rw{false} @Or @Rw{true}}
@producname{exp}@producbody{@bnfNter{Numeral}}
@producname{exp}@producbody{@bnfNter{LiteralString}}
@producname{exp}@producbody{functiondef}
@producname{exp}@producbody{tableconstructor}
@producname{exp}@producbody{@bnfter{...}}
@producname{exp}@producbody{exp
binop exp}
@producname{exp}@producbody{unop exp}
@producname{prefixexp}@producbody{var @Or functioncall @Or
@bnfter{() exp @bnfter{}}}
}
```

Numerals and literal strings are explained in `@See{lexical}`;  
variables are explained in `@See{variables}`;  
function definitions are explained in `@See{func-def}`;  
function calls are explained in `@See{functioncall}`;  
table constructors are explained in `@See{tableconstructor}`.  
Vararg expressions,  
denoted by three dots (`@Char{...}`), can only be used when  
directly inside a variadic function;  
they are explained in `@See{func-def}`.

Binary operators comprise arithmetic operators @see{arith}, bitwise operators @see{bitwise}, relational operators @see{rel-ops}, logical operators @see{logic}, and the concatenation operator @see{concat}.

Unary operators comprise the unary minus @see{arith}, the unary bitwise NOT @see{bitwise}, the unary logical @Rw{not} @see{logic}, and the unary @def{length operator} @see{len-op}.

}

```
@sect3{arith|
@title{Arithmetic Operators}
Lua supports the following @x{arithmetic operators}:
@description{
@item{@T{+}|addition}
@item{@T{-}|subtraction}
@item{@T{*}|multiplication}
@item{@T{/}|float division}
@item{@T{//}|floor division}
@item{@T{%}|modulo}
@item{@T{^}|exponentiation}
@item{@T{-}|unary minus}
}
```

With the exception of exponentiation and float division, the arithmetic operators work as follows:

If both operands are integers, the operation is performed over integers and the result is an integer.

Otherwise, if both operands are numbers, then they are converted to floats, the operation is performed following the machine's rules for floating-point arithmetic (usually the @x{IEEE 754} standard), and the result is a float.

(The string library coerces strings to numbers in arithmetic operations; see @See{coercion} for details.)

Exponentiation and float division (@T{/}) always convert their operands to floats and the result is always a float.

Exponentiation uses the @ANSI{pow}, so that it works for non-integer exponents too.

Floor division (`@T{//}`) is a division that rounds the quotient towards minus infinity, resulting in the floor of the division of its operands.

Modulo is defined as the remainder of a division that rounds the quotient towards minus infinity (floor division).

In case of overflows in integer arithmetic, all operations `@emphx{wrap around}`.  
}

@sect3{bitwise| @title{Bitwise Operators}  
Lua supports the following `@x{bitwise operators}`:  
@description{  
@item{@T{&}}|bitwise AND}  
@item{@T{@VerBar}}|bitwise OR}  
@item{@T{~}}|bitwise exclusive OR}  
@item{@T{>>}}|right shift}  
@item{@T{<<}}|left shift}  
@item{@T{~}}|unary bitwise NOT}  
}

All bitwise operations convert its operands to integers  
`@see{coercion}`,  
operate on all bits of those integers,  
and result in an integer.

Both right and left shifts fill the vacant bits with zeros.  
Negative displacements shift to the other direction;  
displacements with absolute values equal to or higher than  
the  
number of bits in an integer  
result in zero (as all bits are shifted out).  
  
}

@sect3{coercion| @title{Coercions and Conversions}  
Lua provides some automatic conversions between some  
types and representations at run time.  
Bitwise operators always convert float operands to integers.  
Exponentiation and float division  
always convert integer operands to floats.  
All other arithmetic operations applied to mixed numbers  
(integers and floats) convert the integer operand to a float.  
The C API also converts both integers to floats and  
floats to integers, as needed.  
Moreover, string concatenation accepts numbers as arguments,

besides strings.

In a conversion from integer to float,  
if the integer value has an exact representation as a float,  
that is the result.

Otherwise,  
the conversion gets the nearest higher or  
the nearest lower representable value.  
This kind of conversion never fails.

The conversion from float to integer  
checks whether the float has an exact representation as an integer  
(that  
is, the float has an integral value and  
it is in the range of integer representation).  
If it does, that representation is the result.  
Otherwise, the conversion fails.

Several places in Lua coerce strings to numbers when necessary.  
In particular,  
the string library sets metamethods that try to coerce  
strings to numbers in all arithmetic operations.  
If the conversion fails,  
the library calls the metamethod of the other operand  
(if present) or it raises an error.  
Note that bitwise operators do not do this coercion.

It is always a good practice not to rely on the  
implicit coercions from strings to numbers,  
as they are not always applied;  
in particular, `@T{"1"==1}` is false and `@T{"1"<1}` raises an error  
`@see{rel-ops}`.  
These coercions exist mainly for compatibility and may be removed  
in future versions of the language.

A string is converted to an integer or a float  
following its syntax and the rules of the Lua lexer.  
The string may have also leading and trailing whitespaces and a sign.  
All

conversions from strings to numbers  
accept both a dot and the current locale mark  
as the radix character.  
(The Lua lexer, however, accepts only a dot.)  
If the string is not a valid numeral,  
the conversion fails.

If necessary, the result of this first step is then converted  
to a specific number subtype following the previous rules  
for conversions between floats and integers.

The conversion from numbers to strings uses a non-specified human-readable format.

To convert numbers to strings in any specific way, use the function `@Lid{string.format}`.

}

`@sect3{rel-ops| @title{Relational Operators}`

Lua supports the following `@x{relational operators}`:

`@description{`

`@item{@T{==}|equality}`

`@item{@T{~=}|inequality}`

`@item{@T{<}|less than}`

`@item{@T{>}|greater than}`

`@item{@T{<=}|less or equal}`

`@item{@T{>=}|greater or equal}`

`}`

These operators always result in `@false` or `@true`.

Equality (`@T{==}`) first compares the type of its operands.

If the types are different, then the result is `@false`.

Otherwise,

the values of the operands are compared.

Strings are equal if they have the same byte content.

Numbers are equal if they denote the same mathematical value.

Tables, userdata, and threads

are compared by reference:

two objects are considered equal only if they are the same object.

Every time you create a new object

(a table, a userdata, or a thread),

this new object is different from any previously existing object.

A function is always equal to itself.

Functions with any detectable difference

(different behavior, different definition) are always different.

Functions created at different times but with no detectable differences

may be classified as equal or not

(depending on internal caching details).

You can change the way that Lua compares tables and userdata

by using the `@idx{__eq}` metamethod `@see{metatable}`.

Equality comparisons do not convert strings to numbers

or vice versa.

Thus, `@T{"0"]==0}` evaluates to `@false`,

and `@T{t[0]}` and `@T{t["0"]}` denote different

entries in a table.

The operator `@T{~=}` is exactly the negation of equality (`@T{==}`).

The order operators work as follows.

If both arguments are numbers,  
then they are compared according to their mathematical values,  
regardless of their subtypes.

Otherwise, if both arguments are strings,  
then their values are compared according to the current locale.  
Otherwise, Lua tries to call the `@idx{__lt}` or the `@idx{__le}`  
metamethod `@see{metatable}`.

A comparison `@T{a > b}` is translated to `@T{b < a}`  
and `@T{a >= b}` is translated to `@T{b <= a}`.

Following the `@x{IEEE 754}` standard,  
the special value `@x{NaN}` is considered neither less than,  
nor equal to, nor greater than any value, including itself.

}

`@sect3{logic| @title{Logical Operators}`

The `@x{logical operators}` in Lua are

`@Rw{and}`, `@Rw{or}`, and `@Rw{not}`.

Like the control structures `@see{control}`,  
all logical operators consider both `@false` and `@nil` as false  
and anything else as true.

The negation operator `@Rw{not}` always returns `@false` or `@true`.

The conjunction

operator `@Rw{and}` returns its first argument

if this value is `@false` or `@nil`;

otherwise, `@Rw{and}` returns its second argument.

The disjunction operator `@Rw{or}` returns its first argument

if this value is different from `@nil` and `@false`;

otherwise, `@Rw{or}` returns its second argument.

Both `@Rw{and}` and `@Rw{or}` use `@x{short-circuit evaluation}`;

that is,

the second operand is evaluated only if necessary.

Here are some examples:

`@verbatim{`

`10 or 20 --> 10`

`10 or error() --> 10`

`nil or "a" --> "a"`

`nil and 10 --> nil`

`false and error() --> false`

`false and nil --> false`

`false or nil --> nil`

10 and 20 --> 20

}

}

@sect3{concat| @title{Concatenation}

The string `@x{concatenation}` operator in Lua is denoted by two dots (`@Char{..}`).

If both operands are strings or numbers, then the numbers are converted to strings in a non-specified format `@see{coercion}`.

Otherwise, the `@idx{__concat}` metamethod is called `@see{metatable}`.

}

@sect3{len-op|

@title{The Length Operator}

The length operator is denoted by the unary prefix operator `@T{#}`.

The length of a string is its number of bytes.  
(That is the usual meaning of string length when each character is one byte.)

The length operator applied on a table returns a `@x{border}` in that table.

A `@def{border}` in a table `@id{t}` is any non-negative integer that satisfies the following condition:

```
@verbatim{
(border == 0 or t[border] ~= nil) and
(t[border + 1] == nil or border == math.maxinteger)
}
```

In words,

a border is any positive integer index present in the table that is followed by an absent index,  
plus two limit cases:

zero, when index 1 is absent;

and the maximum value for an integer, when that index is present.

Note that keys that are not positive integers do not interfere with borders.

A table with exactly one border is called a `@def{sequence}`.

For instance, the table `@T{{10, 20, 30, 40, 50}}` is a sequence, as it has only one border (5).

The table `@T{{10, 20, 30, nil, 50}}` has two borders (3 and 5),

and therefore it is not a sequence.

(The `@nil` at index 4 is called a `@emphx{hole}`.)

The table `@T{{nil, 20, 30, nil, nil, 60, nil}}`  
 has three borders (0, 3, and 6),  
 so it is not a sequence, too.  
 The table `@T{{}}` is a sequence with border 0.

When `@id{t}` is a sequence,  
`@T{#t}` returns its only border,  
 which corresponds to the intuitive notion of the length of the sequence.  
 When `@id{t}` is not a sequence,  
`@T{#t}` can return any of its borders.  
 (The exact one depends on details of  
 the internal representation of the table,  
 which in turn can depend on how the table was populated and  
 the memory addresses of its non-numeric keys.)

The computation of the length of a table  
 has a guaranteed worst time of `@M{O(log n)}`,  
 where `@M{n}` is the largest integer key in the table.

A program can modify the behavior of the length operator for  
 any value but strings through the `@idx{__len}` metamethod `@see{metatable}`.

}

`@sect3{prec| @title{Precedence}`  
`@x{Operator precedence}` in Lua follows  
 the table below,  
 from lower to higher priority:

```
@verbatim{
or
and
< > <= >= ~= ==
|
~
&
<< >>

..
+ -
* / // %
unary operators (not # - ~)
^
}
```

As usual,  
 you can use parentheses to change the precedences of an expression.  
 The concatenation (`@Char{..}`) and exponentiation (`@Char{^}`)  
 operators are right associative.  
 All other binary operators are left associative.



}

@sect3{tableconstructor| @title{Table Constructors}

Table @x{constructors} are expressions that create tables.

Every time a constructor is evaluated, a new table is created.

A constructor can be used to create an empty table

or to create a table and initialize some of its fields.

The general syntax for constructors is

@Produc{

@producname{tableconstructor}@produciblebody{@bnfter{@Open} @bnfopt{fieldlist} @bnfter{@Close}}

@producname{fieldlist}@produciblebody{field @bnfrep{fieldsep field} @bnfopt{fieldsep}}

@producname{field}@produciblebody{@bnfter{[] exp @bnfter{[]} @bnfter{=} exp @Or

@bnfNter{Name} @bnfter{=} exp @Or exp}

@producname{fieldsep}@produciblebody{@bnfter{,} @Or @bnfter{;}}

}

Each field of the form @T{[exp1] = exp2} adds to the new table an entry with key @id{exp1} and value @id{exp2}.

A field of the form @T{name = exp} is equivalent to

@T{["name"] = exp}.

Fields of the form @id{exp} are equivalent to

@T{[i] = exp}, where @id{i} are consecutive integers starting with 1;

fields in the other formats do not affect this counting.

For example,

@verbatim{

a = { [f(1)] = g; "x", "y"; x = 1, f(x), [30] = 23; 45 }

}

is equivalent to

@verbatim{

do

local t = {}

t[f(1)] = g

t[1] = "x" -- 1st exp

t[2] = "y" -- 2nd exp

t.x = 1 -- t["x"] = 1

t[3] = f(x) -- 3rd exp

t[30] = 23

t[4] = 45 -- 4th exp

a = t

end

}

The order of the assignments in a constructor is undefined.

(This order would be relevant only when there are repeated keys.)

If the last field in the list has the form @id{exp}

and the expression  
is a multires expression,  
then all values returned by this expression enter the list consecutively  
`@see{multires}`.

The field list can have an optional trailing separator,  
as a convenience for machine-generated code.

}

`@sect3{functioncall| @title{Function Calls}`  
A `@x{function call}` in Lua has the following syntax:  
`@Produc{`  
`@producname{functioncall}@producbody{prefixexp args}`  
`}`

In a function call,  
first `@bnfNter{prefixexp}` and `@bnfNter{args}` are evaluated.  
If the value of `@bnfNter{prefixexp}` has type `@emph{function}`,  
then this function is called  
with the given arguments.  
Otherwise, if present,  
the `@bnfNter{prefixexp} @idx{__call}` metamethod is called:  
its first argument is the value of `@bnfNter{prefixexp}`,  
followed by the original call arguments  
`@see{metatable}`.

The form  
`@Produc{`  
`@producname{functioncall}@producbody{prefixexp @bnfter{:} @bnfNter{Name} args}`  
`}`  
can be used to emulate methods.  
A call `@T{v:name(@rep{args})}`  
is syntactic sugar for `@T{v.name(v,@rep{args})}`,  
except that `@id{v}`  
is evaluated only once.

Arguments have the following syntax:  
`@Produc{`  
`@producname{args}@producbody{@bnfter{(} @bnfopt{explist} @bnfter{)}}}`  
`@producname{args}@producbody{tableconstructor}`  
`@producname{args}@producbody{@bnfNter{LiteralString}}`  
`}`

All argument expressions are evaluated before the call.

A call of the form `@T{f{@rep{fields}}}` is  
syntactic sugar for `@T{f({@rep{fields}})}`;  
that is, the argument list is a single new table.

A call of the form `@T{f'@rep{string}'}`  
(or `@T{f"@rep{string}"}` or `@T{f[[@rep{string}]}}`)

is syntactic sugar for `@T{f('@rep{string}')};`  
that is, the argument list is a single literal string.

A call of the form `@T{return @rep{functioncall}}` not in the scope of a to-be-closed variable is called a `@def{tail call}`.

Lua implements `@def{proper tail calls}`

(or `@def{proper tail recursion}`):

In a tail call,

the called function reuses the stack entry of the calling function.

Therefore, there is no limit on the number of nested tail calls that a program can execute.

However,

a tail call erases any debug information about the calling function.

Note that a tail call only happens with a particular syntax, where the `@Rw{return}` has one single function call as argument, and it is outside the scope of any to-be-closed variable.

This syntax makes the calling function return exactly the returns of the called function, without any intervening action.

So, none of the following examples are tail calls:

```
@verbatim{
return (f(x))    -- results adjusted to 1
return 2 * f(x)  -- result multiplied by 2
return x, f(x)   -- additional results
f(x); return    -- results discarded
return x or f(x) -- results adjusted to 1
}

}
```

`@sect3{func-def| @title{Function Definitions}`

The syntax for function definition is

```
@Produc{
@producname{functiondef}@producbody{@Rw{function} funcbody}
@producname{funcbody}@producbody{@bnfter{() @bnfopt{parlist} @bnfter{}} block @Rw{end}}
}
```

The following syntactic sugar simplifies function definitions:

```
@Produc{
@producname{stat}@producbody{@Rw{function}
funcname funcbody}
@producname{stat}@producbody{@Rw{local} @Rw{function} @bnfNter{Name} funcbody}
@producname{funcname}@producbody{@bnfNter{Name} @bnfrep{@bnfter{.} @bnfNter{Name}}}
@bnfopt{@bnfter{:} @bnfNter{Name}}}
}
```

The statement

```
@verbatim{
function f () @rep{body} end
}
```

translates to

```
@verbatim{
f = function () @rep{body} end
}
```

The statement

```
@verbatim{
function t.a.b.c.f () @rep{body} end
}
```

translates to

```
@verbatim{
t.a.b.c.f = function () @rep{body} end
}
```

The statement

```
@verbatim{
local function f () @rep{body} end
}
```

translates to

```
@verbatim{
local f; f = function () @rep{body} end
}
```

not to

```
@verbatim{
local f = function () @rep{body} end
}
```

(This only makes a difference when the body of the function contains references to @id{f}.)

A function definition is an executable expression, whose value has type @emph{function}.

When Lua precompiles a chunk, all its function bodies are precompiled too, but they are not created yet.

Then, whenever Lua executes the function definition, the function is @emph{instantiated} (or @emph{closed}). This function instance, or @emph{closure}, is the final value of the expression.

Parameters act as local variables that are initialized with the argument values:

```
@Produc{
@producname{parlist}@producbody{namelist @bnfopt{@bnfter{,} @bnfter{...}} @Or
@bnfter{...}}
}
```

When a Lua function is called,

it adjusts its list of `@x{arguments}` to the length of its list of parameters `@see{multires}`, unless the function is a `@def{variadic function}`, which is indicated by three dots (`@Char{...}`) at the end of its parameter list.

A variadic function does not adjust its argument list; instead, it collects all extra arguments and supplies them to the function through a `@def{vararg expression}`, which is also written as three dots.

The value of this expression is a list of all actual extra arguments, similar to a function with multiple results `@see{multires}`.

As an example, consider the following definitions:

```
@verbatim{
function
  f(a, b) end
function g(a, b, ...) end
function r() return 1,2,3 end
}
```

Then, we have the following mapping from arguments to parameters and to the vararg expression:

```
@verbatim{
CALL      PARAMETERS

f(3)      a=3, b=nil
f(3, 4)   a=3, b=4
f(3, 4, 5) a=3, b=4
f(r(), 10) a=1, b=10
f(r())    a=1, b=2

g(3)      a=3, b=nil, ... --> (nothing)
g(3, 4)   a=3, b=4,  ... --> (nothing)
g(3, 4, 5, 8) a=3, b=4,  ... --> 5 8
g(5, r())  a=5, b=1,  ... --> 2 3
}
```

Results are returned using the `@Rw{return}` statement `@see{control}`.

If control reaches the end of a function without encountering a `@Rw{return}` statement, then the function returns with no results.

`@index{multiple return}`

There is a system-dependent limit on the number of values that a function may return.

This limit is guaranteed to be greater than 1000.

The `@emphx{colon}` syntax

is used to emulate `@def{methods}`,  
adding an implicit extra parameter `@idx{self}` to  
the function.

Thus, the statement

```
@verbatim{  
function t.a.b.c:f (@rep{params}) @rep{body} end  
}
```

is syntactic sugar for

```
@verbatim{  
t.a.b.c.f = function (self, @rep{params}) @rep{body} end  
}  
  
}
```

`@sect3{multires| @title{Lists of expressions, multiple results,  
and adjustment}`

Both function calls and vararg expressions can result in multiple values.  
These expressions are called `@def{multires expressions}`.

When a multires expression is used as the last element  
of a list of expressions,  
all results from the expression are added to the  
list of values produced by the list of expressions.

Note that a single expression  
in a place that expects a list of expressions  
is the last expression in that (singleton) list.

These are the places where Lua expects a list of expressions:

`@description{`

`@item{A @rw{return} statement,  
for instance @T{return e1, e2, e3} @see{control}.}`

`@item{A table constructor,  
for instance @T{{e1, e2, e3}} @see{tableconstructor}.}`

`@item{The arguments of a function call,  
for  
instance @T{foo(e1, e2, e3)} @see{functioncall}.}`

`@item{A multiple assignment,  
for instance @T{a , b, c = e1, e2, e3} @see{assignment}.}`

`@item{A local declaration,  
for instance @T{local a , b, c = e1, e2, e3} @see{localvar}.}`

`@item{The initial values in a generic @rw{for} loop,`

for instance `@T{for k in e1, e2, e3 do ... end} @see{for}.`

}

In the last four cases,  
the list of values from the list of expressions  
must be `@emph{adjusted}` to a specific length:  
the number of parameters in a call to a non-variadic function  
`@see{func-def}`,  
the number of variables in a multiple assignment or  
a local declaration,  
and exactly four values for a generic `@rw{for}` loop.  
The `@def{adjustment}` follows these rules:  
If there are more values than needed,  
the extra values are thrown away;  
if there are fewer values than needed,  
the list is extended with `@nil`'s.  
When the list of expressions ends with a multiresult expression,  
all results from that expression enter the list of values  
before the adjustment.

When

a multiresult expression is used  
in a list of expressions without being the last element,  
or in a place where the syntax expects a single expression,  
Lua adjusts the result list of that expression to one element.  
As a particular case,  
the syntax expects a single expression inside a parenthesized expression;  
therefore, adding parentheses around a multiresult expression  
forces it to produce exactly one result.

We seldom need to use a variarg expression in a place  
where the syntax expects a single expression.  
(Usually it is simpler to add a regular parameter before  
the variadic part and use that parameter.)  
When there is such a need,  
we recommend assigning the variarg expression  
to a single variable and using that variable  
in its place.

Here are some examples of uses of multiresult expressions.

In all cases, when the construction needs  
`@Q{the n-th result}` and there is no such result,  
it uses a `@nil`.

```
@verbatim{
print(x, f())    -- prints x and all results from f().
print(x, (f())) -- prints
x and the first result from f().
print(f(), x)    -- prints the first result from f() and x.
```

```

print(1 + f())  -- prints 1 added to the first result from f().
local x = ...   -- x gets the first vararg argument.
x,y = ...       -- x gets the first vararg argument,
                -- y gets the second vararg argument.
x,y,z = w, f()  -- x gets w, y gets the first result from f(),
                -- z gets the second result from f().
x,y,z = f()     -- x gets the first result from f(),
                -- y gets the second result from f(),
                -- z gets the third result from f().
x,y,z = f(), g() -- x gets the first result from f(),
                -- y gets the first result from g(),
                -- z gets the second result from g().
x,y,z = (f())   -- x gets the first result from f(), y and z get nil.
return f()     -- returns all results from f().
return x, ...   -- returns x and all received vararg arguments.
return x,y,f()
    -- returns x, y, and all results from f().
{f()}          -- creates a list with all results from f().
{...}          -- creates a list with all vararg arguments.
{f(), 5}       -- creates a list with the first result from f() and 5.
}

}

}

```

@sect2{visibility| @title{Visibility Rules}

@index{visibility}

Lua is a lexically scoped language.

The scope of a local variable begins at the first statement after its declaration and lasts until the last non-void statement of the innermost block that includes the declaration.

(@emph{Void statements} are labels and empty statements.)

Consider the following example:

@verbatim{

```

x = 10          -- global variable
do              -- new block
  local x = x    -- new 'x', with value 10
  print(x)       --> 10
  x = x+1
do              -- another block
  local x = x+1  -- another 'x'
  print(x)       --> 12
end
print(x)        --> 11
end
print(x)        --> 10 (the global one)

```



}

Notice

that, in a declaration like `@T{local x = x}`,  
the new `@id{x}` being declared is not in scope yet,  
and so the second `@id{x}` refers to the outside variable.

Because of the `@x{lexical scoping}` rules,  
local variables can be freely accessed by functions  
defined inside their scope.

A local variable used by an inner function is called an `@def{upvalue}`  
(or `@emphx{external local variable}`, or simply `@emphx{external variable}`)  
inside the inner function.

Notice that each execution of a `@Rw{local}` statement  
defines new local variables.

Consider the following example:

```
@verbatim{
a = {}
local x = 20
for i = 1, 10 do
  local y = 0
  a[i] = function () y = y + 1; return x + y end
end
}
```

The loop creates ten closures  
(that is, ten instances of the anonymous function).  
Each of these closures uses a different `@id{y}` variable,  
while all of them share the same `@id{x}`.

}

}

`@C{-----}`

`@sect1{API| @title{The Application Program Interface}}`

`@simplesect{`

`@index{C`  
`API}`

This section describes the `@N{C API}` for Lua, that is,  
the set of `@N{C functions}` available to the host program to communicate  
with Lua.

All API functions and related types and constants  
are declared in the header file `@defid{lua.h}`.

Even when we use the term @Q{function},  
any facility in the API may be provided as a macro instead.  
Except where stated otherwise,  
all such macros use each of their arguments exactly once  
(except for the first argument, which is always a Lua state),  
and so do not generate any hidden side-effects.

As in most @N{C libraries},  
the Lua API functions do not check their arguments  
for validity or consistency.  
However, you can change this behavior by compiling Lua  
with the macro @defid{LUA\_USE\_APICHECK} defined.

The Lua library is fully reentrant:  
it has no global variables.  
It keeps all information it needs in a dynamic structure,  
called the @def{Lua state}.

Each Lua state has one or more threads,  
which correspond to independent,  
cooperative lines of execution.  
The type @Lid{lua\_State} (despite its name) refers to a thread.  
(Indirectly, through the thread, it also refers to the  
Lua state associated to the thread.)

A pointer to a thread must be passed as the first argument to  
every function in the library, except to @Lid{lua\_newstate},  
which creates a Lua state from scratch and returns a pointer  
to the @emph{main thread} in the new state.

}

@sect2{@title{The Stack}

@simplesect{

Lua uses a @emph{virtual stack} to pass values to and from C.  
Each element in this stack represents a Lua value  
(@nil, number, string, etc.).  
Functions in the API can access this stack through the  
Lua state parameter that they receive.

Whenever Lua calls C, the called function gets a new stack,  
which is independent of previous stacks and of stacks of  
@N{C functions} that are still active.  
This stack initially contains any arguments to the @N{C function}  
and it is where the @N{C function} can store temporary

Lua values and must push  
its results  
to be returned to the caller `@seeC{lua_CFunction}`.

For convenience,  
most query operations in the API do not follow a strict stack discipline.  
Instead, they can refer to any element in the stack  
by using an `@emph{index}:@index{index (API stack)}`  
A positive index represents an absolute stack position,  
starting `@N{at 1}` as the bottom of the stack;  
a negative index represents an offset relative to the top of the stack.  
More specifically, if the stack has `@rep{n}` elements,  
then `@N{index 1}` represents the first element  
(that is, the element that was pushed onto the stack first)  
and  
`@N{index @rep{n}}` represents the last element;  
`@N{index @num{-1}}` also represents the last element  
(that is, the element at `@N{the top}`)  
and index `@M{-n}` represents the first element.

}

`@sect3{stacksize| @title{Stack Size}`

When you interact with the Lua API,  
you are responsible for ensuring consistency.  
In particular,  
`@emph{you are responsible for controlling stack overflow}`.  
When you call any  
API function,  
you must ensure the stack has enough room to accommodate the results.

There is one exception to the above rule:  
When you call a Lua function  
without a fixed number of results `@seeF{lua_call}`,  
Lua ensures that the stack has enough space for all results.  
However, it does not ensure any extra space.  
So, before pushing anything on the stack after such a call  
you should use `@Lid{lua_checkstack}`.

Whenever Lua calls C,  
it ensures that the stack has space for  
at least `@defid{LUA_MINSTACK}` extra elements;  
that is, you can safely push up to `@id{LUA_MINSTACK}` values into it.  
`@id{LUA_MINSTACK}` is defined as 20,  
so that usually you do not have to worry about stack space  
unless your code has loops pushing elements onto the stack.  
Whenever necessary,

you can use the function `@Lid{lua_checkstack}`  
to ensure that the stack has enough space for pushing new elements.

}

`@sect3{@title{Valid and Acceptable Indices}`

Any function in the API that receives stack indices  
works only with `@emphx{valid`  
`indices}` or `@emphx{acceptable indices}`.

A `@def{valid index}` is an index that refers to a  
position that stores a modifiable Lua value.  
It comprises stack indices `@N{between 1}` and the stack top  
(`@T{1 @leq abs(index) @leq top}`)  
`@index{stack index}`  
plus `@def{pseudo-indices}`,  
which represent some positions that are accessible to `@N{C code}`  
but that are not in the stack.  
Pseudo-indices are used to access the registry `@see{registry}`  
and the upvalues of a `@N{C function}` `@see{c-closure}`.

Functions that do not need a specific mutable position,  
but only a value (e.g., query functions),  
can be called with acceptable indices.  
An `@def{acceptable index}` can be any valid index,  
but it also can be any positive index after the stack top  
within the space allocated for the stack,  
that is, indices up to the stack size.  
(Note that 0 is never an acceptable index.)  
Indices to upvalues `@see{c-closure}` greater than the real number  
of upvalues in the current `@N{C function}` are also acceptable (but invalid).  
Except  
when noted otherwise,  
functions in the API work with acceptable indices.

Acceptable indices serve to avoid extra tests  
against the stack top when querying the stack.  
For instance, a `@N{C function}` can query its third argument  
without the need to check whether there is a third argument,  
that is, without the need to check whether 3 is a valid index.

For functions that can be called with acceptable indices,  
any non-valid index is treated as if it  
contains a value of a virtual type `@defid{LUA_TNONE}`,  
which behaves like a nil value.

}

@sect3{constchar|@title{Pointers to strings}

Several functions in the API return pointers ( $@T\{\text{const char}^*\}$ ) to Lua strings in the stack.

(See @Lid{lua\_pushfstring}, @Lid{lua\_pushlstring},

@Lid{lua\_pushstring}, and @Lid{lua\_tolstring}.

See also @Lid{luaL\_checklstring}, @Lid{luaL\_checkstring}, and @Lid{luaL\_tolstring} in the auxiliary library.)

In general,

Lua's garbage collection can free or move internal memory and then invalidate pointers to internal strings.

To allow a safe use of these pointers, the API guarantees that any pointer to a string in a stack index is valid while the string value at that index is not removed from the stack. (It can be moved to another index, though.)

When the index is a pseudo-index (referring to an upvalue), the pointer is valid while the corresponding call is active and the corresponding upvalue is not modified.

Some functions in the debug interface also return pointers to strings, namely @Lid{lua\_getlocal}, @Lid{lua\_getupvalue}, @Lid{lua\_setlocal}, and @Lid{lua\_setupvalue}. For these functions, the pointer is guaranteed to be valid while the caller function is active and the given closure (if one was given) is in the stack.

Except for these guarantees, the garbage collector is free to invalidate any pointer to internal strings.

}

}

@sect2{c-closure| @title{C Closures}

When a  $@N\{\text{C function}\}$  is created, it is possible to associate some values with it, thus creating a  $@def\{ @N\{\text{C closure}\} \}$  @seeC{lua\_pushcclosure}; these values are called  $@def\{\text{upvalues}\}$  and are accessible to the function whenever it is called.

Whenever a `@N{C function}` is called,  
 its upvalues are located at specific pseudo-indices.  
 These pseudo-indices are produced by the macro  
`@Lid{lua_upvalueindex}`.  
 The first upvalue associated with a function is at index  
`@T{lua_upvalueindex(1)}`, and so on.  
 Any access to `@T{lua_upvalueindex(@rep{n})}`,  
 where `@rep{n}` is greater than the number of upvalues of the  
 current function  
 (but not greater than 256,  
 which is one plus the maximum number of upvalues in a closure),  
 produces an acceptable but invalid index.

A `@N{C closure}` can also change the values  
 of its corresponding upvalues.

}

`@sect2{registry| @title{Registry}}`

Lua provides a `@def{registry}`,  
 a predefined table that can be used by any `@N{C code}` to  
 store whatever Lua values it needs to store.  
 The registry table is always accessible at pseudo-index  
`@defid{LUA_REGISTRYINDEX}`.  
 Any `@N{C library}` can store data into  
 this table,  
 but it must take care to choose keys  
 that are different from those used  
 by other libraries, to avoid collisions.  
 Typically, you should use as key a string containing your library name,  
 or a light userdata with the address of a `@N{C object}` in your code,  
 or any Lua object created by your code.  
 As with variable names,  
 string keys starting with an underscore followed by  
 uppercase letters are reserved for Lua.

The integer keys in the registry are used  
 by the reference mechanism `@seeC{luaL_ref}`  
 and by some predefined values.  
 Therefore, integer keys in the registry  
 must not be used for other purposes.

When you create a new Lua state,  
 its registry comes with some predefined values.  
 These predefined values are indexed with integer keys  
 defined as constants in `@id{lua.h}`.  
 The following constants are defined:

```
@description{
@item{@defid{LUA_RIDX_MAINTHREAD}}| At this index the registry has
the main thread of the state.
(The main thread is the one created together with the state.)
}
```

```
@item{@defid{LUA_RIDX_GLOBALS}}|
  At this index the registry has
the @x{global environment}.
}
}
}
```

```
@sect2{C-error|@title{Error Handling in C}}
```

```
@simplesect{
```

Internally, Lua uses the C @id{longjmp} facility to handle errors.  
 (Lua will use exceptions if you compile it as C++;  
 search for @id{LUA\_THROW} in the source code for details.)  
 When Lua faces any error,  
 such as a @x{memory allocation error} or a type error,  
 it @emph{raises} an error;  
 that is, it does a long jump.  
 A @emphx{protected environment} uses @id{setjmp}  
 to set a recovery point;  
 any error jumps to the most recent active recovery point.

Inside a @N{C function} you can raise an error explicitly  
 by calling @Lid{lua\_error}.

Most functions in the API can raise an error,  
 for instance due to a @x{memory allocation error}.  
 The documentation for each function indicates whether  
 it can raise errors.

If an error happens outside any protected environment,  
 Lua calls a @def{panic function} (see @Lid{lua\_atpanic})  
 and then calls @T{abort},  
 thus  
 exiting the host application.  
 Your panic function can avoid this exit by  
 never returning  
 (e.g., doing a long jump to your own recovery point outside Lua).

The panic function,  
 as its name implies,

is a mechanism of last resort.

Programs should avoid it.

As a general rule,

when a `@N{C function}` is called by Lua with a Lua state,

it can do whatever it wants on that Lua state,

as it should be already protected.

However,

when C code operates on other Lua states

(e.g., a Lua-state argument to the function,

a Lua state stored in the registry, or

the result of `@Lid{lua_newthread}`),

it should use them only in API calls that cannot raise errors.

The panic function runs as if it were a `@x{message handler} @see{error}`;

in particular, the error object is on the top of the stack.

However, there is no guarantee about stack space.

To push anything on the stack,

the panic function must first check the available space `@see{stacksize}`.

}

`@sect3{statuscodes|@title{Status Codes}`

Several functions

that report errors in the API use the following

status codes to indicate different kinds of errors or other conditions:

`@description{`

`@item{@defid{LUA_OK} (0)| no errors.}`

`@item{@defid{LUA_ERRRUN}| a runtime error.}`

`@item{@defid{LUA_ERRMEM}|`

`@x{memory allocation error}.`

For such errors, Lua does not call the `@x{message handler}`.

}

`@item{@defid{LUA_ERRERR}| error while running the @x{message handler}.}`

`@item{@defid{LUA_ERRSYNTAX}| syntax error during precompilation.}`

`@item{@defid{LUA_YIELD}| the thread (coroutine) yields.}`

`@item{@defid{LUA_ERRFILE}| a file-related error;`

e.g., it cannot open or read the file.}

}



These constants are defined in the header file @id{lua.h}.

```
}  
  
}
```

@sect2{continuations|@title{Handling Yields in C}}

Internally, Lua uses the C @id{longjmp} facility to yield a coroutine. Therefore, if a @N{C function} @id{foo} calls an API function and this API function yields (directly or indirectly by calling another function that yields), Lua cannot return to @id{foo} any more, because the @id{longjmp} removes its frame from the @N{C stack}.

To avoid this kind of problem, Lua raises an error whenever it tries to yield across an API call, except for three functions: @Lid{lua\_yieldk}, @Lid{lua\_callk}, and @Lid{lua\_pcallk}. All those functions receive a @def{continuation function} (as a parameter named @id{k}) to continue execution after a yield.

We need to set some terminology to explain continuations. We have a @N{C function} called from Lua which we will call the @emph{original function}. This original function then calls one of those three functions in the C API, which we will call the @emph{callee function}, that then yields the current thread. This can happen when the callee function is @Lid{lua\_yieldk}, or when the callee function is either @Lid{lua\_callk} or @Lid{lua\_pcallk} and the function called by them yields.

Suppose the running thread yields while executing the callee function. After the thread resumes, it eventually will finish running the callee function. However, the callee function cannot return to the original function, because its frame in the @N{C stack} was destroyed by the yield. Instead, Lua calls a @def{continuation function}, which was given as an argument to the callee function. As the name implies, the continuation function should continue the task of the original function.

As an illustration, consider the following function:

@verbatim{

```

int original_function (lua_State *L) {
...   /* code 1 */
status = lua_pcall(L, n, m, h); /* calls Lua */
...   /* code 2 */
}
}

```

Now we want to allow

the Lua code being run by @Lid{lua\_pcall} to yield.

First, we can rewrite our function like here:

```

@verbatim{
int k (lua_State *L, int status, lua_KContext ctx) {
...   /* code 2 */
}
}

```

```

int original_function (lua_State *L) {
...   /* code 1 */
return k(L, lua_pcall(L, n, m, h), ctx);
}
}

```

In the above code,

the new function @id{k} is a

@emph{continuation function} (with type @Lid{lua\_KFunction}),  
which

should do all the work that the original function  
was doing after calling @Lid{lua\_pcall}.

Now, we must inform Lua that it must call @id{k} if the Lua code  
being executed by @Lid{lua\_pcall} gets interrupted in some way  
(errors or yielding),

so we rewrite the code as here,

replacing @Lid{lua\_pcall} by @Lid{lua\_pcallk}:

```

@verbatim{
int original_function (lua_State *L) {
...   /* code 1 */
return k(L, lua_pcallk(L, n, m, h, ctx2, k), ctx1);
}
}

```

Note the external, explicit call to the continuation:

Lua will call the continuation only if needed, that is,

in case of errors or resuming after a yield.

If the called function returns normally without ever yielding,

@Lid{lua\_pcallk} (and @Lid{lua\_callk}) will also return normally.

(Of course, instead of calling the continuation in that case,

you can do the equivalent work directly inside the original function.)

Besides the Lua state,

the continuation function has two other parameters:

the final status of the call and the context value

(@id{ctx}) that  
 was passed originally to @Lid{lua\_pcallk}.  
 Lua does not use this context value;  
 it only passes this value from the original function to the  
 continuation function.  
 For @Lid{lua\_pcallk},  
 the status is the same value that would be returned by @Lid{lua\_pcallk},  
 except that it is @Lid{LUA\_YIELD} when being executed after a yield  
 (instead of @Lid{LUA\_OK}).  
 For @Lid{lua\_yieldk} and @Lid{lua\_callk},  
 the status is always @Lid{LUA\_YIELD} when Lua calls the continuation.  
 (For these two functions,  
 Lua will not call the continuation in case of errors,  
 because they do not handle errors.)  
 Similarly, when using @Lid{lua\_callk},  
 you should call the continuation function  
 with @Lid{LUA\_OK} as the status.  
 (For @Lid{lua\_yieldk}, there is not much point in calling  
 directly the continuation function,  
 because @Lid{lua\_yieldk} usually does not return.)

Lua treats the continuation function as if it were the original function.  
 The continuation function receives the same Lua stack  
 from the original  
 function,  
 in the same state it would be if the callee function had returned.  
 (For instance,  
 after a @Lid{lua\_callk} the function and its arguments are  
 removed from the stack and replaced by the results from the call.)  
 It also has the same upvalues.  
 Whatever it returns is handled by Lua as if it were the return  
 of the original function.

}

@sect2{@title{Functions and Types}}

Here we list all functions and types from the @N{C API} in  
 alphabetical order.

Each function has an indicator like this:

@apii{o,p,x}

The first field, @T{o},  
 is how many elements the function pops from the stack.  
 The second field, @T{p},  
 is how many elements the function pushes onto the stack.  
 (Any function always pushes its results after popping its arguments.)  
 A field in the form @T{x|y} means the function can push (or pop)

@T{x} or @T{y} elements,  
depending on the situation;  
an interrogation mark @Char{?} means that  
we cannot know how many elements the function pops/pushes  
by looking only at its arguments.

(For

instance, they may depend on what is in the stack.)

The third field, @T{x},

tells whether the function may raise errors:

@Char{-} means the function never raises any error;

@Char{m} means the function may raise only out-of-memory errors;

@Char{v} means the function may raise the errors explained in the text;

@Char{e} means the function can run arbitrary Lua code,

either directly or through metamethods,

and therefore may raise any errors.

```
@APIEntry{int lua_absindex (lua_State *L, int idx);|
```

```
@apii{0,0,-}
```

Converts the @x{acceptable index} @id{idx}

into an equivalent @x{absolute index}

(that is, one that does not depend on the stack size).

```
}
```

```
@APIEntry{
```

```
typedef void * (*lua_Alloc) (void *ud,
```

```
void *ptr,
```

```
size_t osize,
```

```
size_t nsize);|
```

The type of the @x{memory-allocation function} used by Lua states.

The allocator function must provide a

functionality similar to @id{realloc},

but not exactly the

same.

Its arguments are

@id{ud}, an opaque pointer passed to @Lid{lua\_newstate};

@id{ptr}, a pointer to the block being allocated/reallocated/freed;

@id{osize}, the original size of the block or some code about what  
is being allocated;

and @id{nsize}, the new size of the block.

When @id{ptr} is not @id{NULL},

@id{osize} is the size of the block pointed by @id{ptr},

that is, the size given when it was allocated or reallocated.

When @id{ptr} is @id{NULL},  
 @id{osize} encodes the kind of object that Lua is allocating.  
 @id{osize} is any of  
 @Lid{LUA\_TSTRING}, @Lid{LUA\_TTABLE}, @Lid{LUA\_TFUNCTION},  
 @Lid{LUA\_TUSERDATA}, or @Lid{LUA\_TTHREAD} when (and only when)  
 Lua is creating a new object of that type.  
 When @id{osize} is some other value,  
 Lua is allocating memory for something else.

Lua assumes the following behavior from the allocator function:

When @id{nsize} is zero,  
 the allocator must behave like @id{free}  
 and then return @id{NULL}.

When @id{nsize} is not zero,  
 the allocator must behave  
 like @id{realloc}.  
 In particular, the allocator returns @id{NULL}  
 if and only if it cannot fulfill the request.

Here is a simple implementation for the @x{allocator function}.  
 It is used in the auxiliary library by @Lid{luaL\_newstate}.

```
@verbatim{
static void *l_alloc (void *ud, void *ptr, size_t osize,
                      size_t nsize) {
    (void)ud; (void)osize; /* not used */
    if (nsize == 0) {
        free(ptr);
        return NULL;
    }
    else
        return realloc(ptr, nsize);
}
}
```

Note that @N{ISO C} ensures  
 that @T{free(NULL)} has no effect and that  
 @T{realloc(NULL,size)} is equivalent to @T{malloc(size)}.

```
@APIEntry{void lua_arith (lua_State *L, int op);|
@apii{2|1,1,e}
```

Performs an arithmetic or bitwise operation over the two values  
 (or one, in the case of negations)  
 at the top of the stack,

with the value on the top being the second operand,  
pops these values, and pushes the result of the operation.  
The function follows the semantics of the corresponding Lua  
operator  
(that is, it may call metamethods).

The value of `@id{op}` must be one of the following constants:

`@description{`

```
@item{@defid{LUA_OPADD}| performs addition (@T{+})}
@item{@defid{LUA_OPSUB}| performs subtraction (@T{-})}
@item{@defid{LUA_OPMUL}| performs multiplication (@T{*})}
@item{@defid{LUA_OPDIV}| performs float division (@T{/})}
@item{@defid{LUA_OPIDIV}| performs floor division (@T{//})}
@item{@defid{LUA_OPMOD}| performs modulo (@T{%})}
@item{@defid{LUA_OPPOW}| performs exponentiation (@T{^})}
@item{@defid{LUA_OPUNM}| performs mathematical negation (unary @T{-})}
@item{@defid{LUA_OPBNOT}| performs bitwise NOT (@T{~})}
@item{@defid{LUA_OPBAND}| performs bitwise AND (@T{&})}
@item{@defid{LUA_OPBOR}| performs bitwise OR (@T{|})}
@item{@defid{LUA_OPBXOR}| performs bitwise exclusive OR (@T{~})}
@item{@defid{LUA_OPSHL}| performs left shift (@T{<<})}
@item{@defid{LUA_OPSHR}| performs right shift (@T{>>})}
```

}

}

```
@APIEntry{lua_CFunction lua_atpanic (lua_State *L, lua_CFunction
panicf);|
@apii{0,0,-}
```

Sets a new panic function and returns the old one @see{C-error}.

}

```
@APIEntry{void lua_call (lua_State *L, int nargs, int nresults);|
@apii{nargs+1,nresults,e}
```

Calls a function.

Like regular Lua calls,

`@id{lua_call}` respects the `@idx{__call}` metamethod.

So, here the word `@Q{function}`

means any callable value.

To do a call you must use the following protocol:

first, the function to be called is pushed onto the stack;

then, the arguments to the call are pushed

in direct order;  
that is, the first argument is pushed first.  
Finally you call `@Lid{lua_call};`  
`@id{nargs}` is the number of arguments that you pushed onto the stack.  
When the function returns,  
all arguments and the function value are popped  
and the call results are pushed onto the stack.  
The number of results is adjusted to `@id{nresults}`,  
unless `@id{nresults}` is `@defid{LUA_MULTRET}`.  
In this case, all results from the function are pushed;  
Lua takes care that the returned values fit into the stack space,  
but  
it does not ensure any extra space in the stack.  
The function results are pushed onto the stack in direct order  
(the first result is pushed first),  
so that after the call the last result is on the top of the stack.

Any error while calling and running the function is propagated upwards  
(with a `@id{longjmp}`).

The following example shows how the host program can do the  
equivalent to this Lua code:

```
@verbatim{
a = f("how", t.x, 14)
}

Here it is @N{in C}:
@verbatim{
lua_getglobal(L, "f");           /* function to be called */
lua_pushliteral(L, "how");       /* 1st argument */
lua_getglobal(L, "t");           /* table to be indexed */
lua_getfield(L, -1, "x");        /* push result of t.x (2nd arg) */
lua_remove(L, -2);              /* remove 't' from the stack */
lua_pushinteger(L, 14);          /* 3rd argument */
lua_call(L, 3, 1);              /* call 'f' with 3 arguments and 1 result */
lua_setglobal(L, "a");           /* set global
'a' */
}
```

Note that the code above is `@emph{balanced}`:  
at its end, the stack is back to its original configuration.  
This is considered good programming practice.

}

```
@APIEntry{
void lua_callk (lua_State *L,
               int nargs,
               int nresults,
               lua_KContext ctx,
```

```
lua_KFunction k);|
@apii{nargs + 1,nresults,e}
```

This function behaves exactly like @Lid{lua\_call},  
but allows the called function to yield @see{continuations}.

```
}
```

```
@APIEntry{typedef int (*lua_CFunction) (lua_State *L);|
```

Type for @N{C functions}.

In order to communicate properly with Lua,  
a @N{C function} must use the following protocol,  
which defines the way parameters and results are passed:  
a @N{C function} receives its arguments from Lua in its stack  
in direct order (the first argument is pushed first).  
So, when the function starts,  
@T{lua\_gettop(L)} returns the number of arguments received by the function.  
The first argument (if any) is at index 1  
and its last argument  
is at index @T{lua\_gettop(L)}.  
To return values to Lua, a @N{C function} just pushes them onto the stack,  
in direct order (the first result is pushed first),  
and returns in C the number of results.  
Any other value in the stack below the results will be properly  
discarded by Lua.  
Like a Lua function, a @N{C function} called by Lua can also return  
many results.

As an example, the following function receives a variable number  
of numeric arguments and returns their average and their sum:

```
@verbatim{
static int foo (lua_State *L) {
  int n = lua_gettop(L);  /* number of arguments */
  lua_Number sum = 0.0;
  int i;
  for (i = 1; i <= n; i++) {
    if (!lua_isnumber(L, i)) {
      lua_pushliteral(L, "incorrect argument");
      lua_error(L);
    }
    sum += lua_tonumber(L, i);
  }
  lua_pushnumber(L, sum/n);  /* first result */
  lua_pushnumber(L, sum);   /* second result */
  return 2;                 /* number of results */
}
```



```
}
```

```
}
```

```
@APIEntry{int lua_checkstack (lua_State  
*L, int n);|  
@apii{0,0,-}}
```

Ensures that the stack has space for at least @id{n} extra elements, that is, that you can safely push up to @id{n} values into it. It returns false if it cannot fulfill the request, either because it would cause the stack to be greater than a fixed maximum size (typically at least several thousand elements) or because it cannot allocate memory for the extra space. This function never shrinks the stack; if the stack already has space for the extra elements, it is left unchanged.

```
}
```

```
@APIEntry{void lua_close (lua_State *L);|  
@apii{0,0,-}}
```

Close all active to-be-closed variables in the main thread, release all objects in the given Lua state (calling the corresponding garbage-collection metamethods, if any), and frees all dynamic memory used by this state.

On several platforms, you may not need to call this function, because all resources are naturally released when the host program ends. On the other hand, long-running programs that create multiple states, such as daemons or web servers, will probably need to close states as soon as they are not needed.

```
}
```

```
@APIEntry{void lua_closeslot (lua_State *L, int index);|  
@apii{0,0,e}}
```

Close the to-be-closed slot at the given index and set its value to @nil. The index must be the last index previously marked to be closed @see{lua\_toclose} that is still active (that is, not closed yet).

A @idx{\_\_close} metamethod cannot yield when called through this function.

(This function was introduced in @N{release 5.4.3}.)

}

@APIEntry{int lua\_closethread (lua\_State \*L, lua\_State \*from);|  
@apii{0,?,~}}

Resets a thread, cleaning its call stack and closing all pending to-be-closed variables.

Returns a status code:

@Lid{LUA\_OK} for no errors in the thread (either the original error that stopped the thread or errors in closing methods),  
or an error status otherwise.

In case of error,

leaves the error object on the top of the stack.

The parameter @id{from} represents the coroutine that is resetting @id{L}.

If

there is no such coroutine,  
this parameter can be @id{NULL}.

(This function was introduced in @N{release 5.4.6}.)

}

@APIEntry{int lua\_compare (lua\_State \*L, int index1, int index2, int op);|  
@apii{0,0,e}}

Compares two Lua values.

Returns 1 if the value at index @id{index1} satisfies @id{op} when compared with the value at index @id{index2}, following the semantics of the corresponding Lua operator (that is, it may call metamethods).

Otherwise @N{returns 0}.

Also @N{returns 0} if any of the indices is not valid.

The value of @id{op} must be one of the following constants:

@description{

@item{@defid{LUA\_OPEQ}| compares for equality (@T{==})}

@item{@defid{LUA\_OPLT}| compares for less than (@T{<})}

@item{@defid{LUA\_OPLE}| compares for less or equal (@T{<=})}

}

```
}
```

```
@APIEntry{void lua_concat (lua_State *L, int n);|  
@apii{n,1,e}}
```

Concatenates the @id{n} values at the top of the stack,  
pops them, and leaves the result on the top.

If @N{@T{n} is 1}, the result is the single value on the  
stack

(that is, the function does nothing);

if @id{n} is 0, the result is the empty string.

Concatenation is performed following the usual semantics of Lua  
@see{concat}.

```
}
```

```
@APIEntry{void lua_copy (lua_State *L, int fromidx, int toidx);|  
@apii{0,0,-}}
```

Copies the element at index @id{fromidx}  
into the valid index @id{toidx},  
replacing the value at that position.

Values at other positions are not affected.

```
}
```

```
@APIEntry{void lua_createtable (lua_State *L, int narr, int nrec);|  
@apii{0,1,m}}
```

Creates a new empty table and pushes it onto the stack.

Parameter @id{narr} is a hint for how many elements the table  
will have as a sequence;

parameter @id{nrec} is a hint for how many other elements  
the table will have.

Lua may use these hints to preallocate memory for the new table.

This preallocation may help performance when you know in advance  
how many elements the table will have.

Otherwise you can use the function @Lid{lua\_newtable}.

```
}
```

```
@APIEntry{int lua_dump (lua_State *L,
```

```
    lua_Writer writer,  
    void *data,  
    int strip);|
```

```
@apii{0,0,-}}
```

Dumps a function as a binary chunk.  
Receives a Lua function on the top of the stack  
and produces a binary chunk that,  
if loaded again,  
results in a function equivalent to the one dumped.  
As it produces parts of the chunk,  
@Lid{lua\_dump} calls function @id{writer} @seeC{lua\_Writer}  
with the given @id{data}  
to write them.

If @id{strip} is true,  
the binary representation may not include all debug information  
about the function,  
to save space.

The value returned is the error code returned by the last  
call to the writer;  
@N{0 means} no errors.

This function does not pop the Lua function from the stack.

}

@APIEntry{int lua\_error (lua\_State \*L);|  
@apii{1,0,v}}

Raises a Lua error,  
using the value on the top of the stack as the error object.  
This function does a long jump,  
and therefore never returns  
@seeC{luaL\_error}.

}

@APIEntry{int lua\_gc (lua\_State \*L, int what, ...);|  
@apii{0,0,-}}

Controls  
the garbage collector.

This function performs several tasks,  
according to the value of the parameter @id{what}.  
For options that need extra arguments,  
they are listed after the option.  
@description{

@item{@id{LUA\_GCCOLLECT}|

Performs a full garbage-collection cycle.

}

@item{@id{LUA\_GCSTOP}|

Stops the garbage collector.

}

@item{@id{LUA\_GCRESTART}|

Restarts the garbage collector.

}

@item{@id{LUA\_GCCOUNT}|

Returns the current amount of memory (in Kbytes) in use by Lua.

}

@item{@id{LUA\_GCCOUNTB}|

Returns the remainder of dividing the current amount of bytes of memory in use by Lua by 1024.

}

@item{@id{LUA\_GCSTEP} @T{(int stepsize)}|

Performs an incremental step of garbage collection, corresponding to the allocation of @id{stepsize} Kbytes.

}

@item{@id{LUA\_GCISRUNNING}|

Returns a boolean that tells whether the collector is running (i.e., not stopped).

}

@item{@id{LUA\_GCINC} (int pause, int stepmul, stepsize)|

Changes the collector to incremental mode with the

given parameters @see{incmode}.

Returns the previous mode (@id{LUA\_GCGEN} or @id{LUA\_GCINC}).

}

@item{@id{LUA\_GCGEN} (int minormul, int majormul)|

Changes the collector to generational mode with the given parameters @see{genmode}.

Returns the previous mode (@id{LUA\_GCGEN} or @id{LUA\_GCINC}).

}

}

For more details about these options, see @Lid{collectgarbage}.

This function should not be called by a finalizer.

}

```
@APIEntry{lua_Alloc lua_getallocf (lua_State *L, void **ud);|
@apii{0,0,-}
```

Returns the @x{memory-allocation function} of a given state.  
If @id{ud} is not @id{NULL}, Lua stores in @T{\*ud} the  
opaque pointer given when the memory-allocator function was set.

}

```
@APIEntry{int lua_getfield (lua_State *L, int index, const char *k);|
@apii{0,1,e}
```

Pushes onto the stack the value @T{t[k]},  
where @id{t} is the value at the given index.  
As in Lua, this function may trigger a metamethod  
for the @Q{index} event @see{metatable}.

Returns the type of the pushed value.

}

```
@APIEntry{void
*lua_getextraspace (lua_State *L);|
@apii{0,0,-}
```

Returns a pointer to a raw memory area associated with the  
given Lua state.  
The application can use this area for any purpose;  
Lua does not use it for anything.

Each new thread has this area initialized with a copy  
of the area of the @x{main thread}.

By default, this area has the size of a pointer to void,  
but you can recompile Lua with a different size for this area.  
(See @id{LUA\_EXTRASPACE} in @id{luaconf.h}.)

}

```
@APIEntry{int lua_getglobal (lua_State *L, const char *name);|
@apii{0,1,e}
```

Pushes onto the stack the value of the global @id{name}.  
Returns the type of that value.

}

```
@APIEntry{int lua_geti (lua_State *L, int index, lua_Integer i);|
@apii{0,1,e}
```

Pushes onto the stack the value `@T{t[i]}`,  
where `@id{t}` is the value at the given index.  
As in Lua, this function may trigger a metamethod  
for the `@Q{index}` event `@see{metatable}`.

Returns the type of the pushed value.

}

```
@APIEntry{int lua_getmetatable (lua_State *L, int index);|
@apii{0,0|1,-}
```

If  
the value at the given index has a metatable,  
the function pushes that metatable onto the stack and `@N{returns 1}`.  
Otherwise,  
the function `@N{returns 0}` and pushes nothing on the stack.

}

```
@APIEntry{int lua_gettable (lua_State *L, int index);|
@apii{1,1,e}
```

Pushes onto the stack the value `@T{t[k]}`,  
where `@id{t}` is the value at the given index  
and `@id{k}` is the value on the top of the stack.

This function pops the key from the stack,  
pushing the resulting value in its place.  
As in Lua, this function may trigger a metamethod  
for the `@Q{index}` event `@see{metatable}`.

Returns the type of the pushed value.

}

```
@APIEntry{int lua_gettop (lua_State *L);|
@apii{0,0,-}
```

Returns the index of the top element in the stack.  
Because indices start `@N{at 1}`,  
this result is equal to the number of elements in the stack;  
in particular, `@N{0}` means an empty stack.

```
}
```

```
@APIEntry{int lua_getiuservalue (lua_State *L, int index, int n);|  
@apii{0,1,-}}
```

Pushes onto the stack the @id{n}-th  
user value associated with the  
full userdata at the given index and  
returns the type of the pushed value.

If the userdata does not have that value,  
pushes @nil and returns @Lid{LUA\_TNONE}.

```
}
```

```
@APIEntry{void lua_insert (lua_State *L, int index);|  
@apii{1,1,-}}
```

Moves the top element into the given valid index,  
shifting up the elements above this index to open space.  
This function cannot be called with a pseudo-index,  
because a pseudo-index is not an actual stack position.

```
}
```

```
@APIEntry{typedef @ldots lua_Integer;|
```

The type of integers in Lua.

By default this type is @id{long long},  
(usually a 64-bit two-complement integer),  
but that can be changed to @id{long} or @id{int}  
(usually a 32-bit two-complement integer).  
(See @id{LUA\_INT\_TYPE} in @id{luaconf.h}.)

Lua also defines the constants

@defid{LUA\_MININTEGER} and @defid{LUA\_MAXINTEGER},  
with the minimum and the maximum values that fit in this type.

```
}
```

```
@APIEntry{int lua_isboolean (lua_State *L, int index);|  
@apii{0,0,-}}
```

Returns 1  
if the value at the given index is a boolean,  
and @N{0 otherwise}.



```
}
```

```
@APIEntry{int lua_iscfunction (lua_State *L, int index);|  
@apii{0,0,-}}
```

Returns 1 if the value at the given index is a @N{C function},  
and @N{0 otherwise}.

```
}
```

```
@APIEntry{int lua_isfunction (lua_State *L, int index);|  
@apii{0,0,-}}
```

Returns 1 if the value at the given index is a function  
(either C or Lua), and @N{0 otherwise}.

```
}
```

```
@APIEntry{int lua_isinteger (lua_State *L, int index);|  
@apii{0,0,-}}
```

Returns 1 if the value at the given index is an integer  
(that is, the value is a number and is represented as an integer),  
and @N{0 otherwise}.

```
}
```

```
@APIEntry{int lua_islightuserdata (lua_State *L, int index);|  
@apii{0,0,-}}
```

Returns 1 if the value at the given index is a light userdata,  
and @N{0 otherwise}.

```
}
```

```
@APIEntry{int lua_isnil (lua_State *L, int index);|  
@apii{0,0,-}}
```

Returns 1 if the value at the given index is @nil,  
and @N{0 otherwise}.

```
}
```

```
@APIEntry{int lua_isnone (lua_State *L, int index);|  
@apii{0,0,-}}
```

Returns

1 if the given index is not valid,  
and @N{0 otherwise}.

}

@APIEntry{int lua\_isnoneornil (lua\_State \*L, int index);|  
@apii{0,0,-}}

Returns 1 if the given index is not valid  
or if the value at this index is @nil,  
and @N{0 otherwise}.

}

@APIEntry{int lua\_isnumber (lua\_State \*L, int index);|  
@apii{0,0,-}}

Returns 1 if the value at the given index is a number  
or a string convertible to a number,  
and @N{0 otherwise}.

}

@APIEntry{int lua\_isstring (lua\_State \*L, int index);|  
@apii{0,0,-}}

Returns 1 if the value at the given index is a string  
or a number (which is always convertible to a string),  
and @N{0 otherwise}.

}

@APIEntry{int lua\_istable (lua\_State \*L, int index);|  
@apii{0,0,-}}

Returns 1 if the value at the given index is a table,  
and @N{0 otherwise}.

}

@APIEntry{int lua\_isthread (lua\_State \*L, int index);|  
@apii{0,0,-}}

Returns 1 if the value at the given index is a thread,  
and @N{0 otherwise}.

}

```
@APIEntry{int lua_isuserdata (lua_State *L, int index);|
@apii{0,0,-}}
```

Returns

1 if the value at the given index is a userdata  
(either full or light), and @N{0 otherwise}.

```
}
```

```
@APIEntry{int lua_isyieldable (lua_State *L);|
@apii{0,0,-}}
```

Returns 1 if the given coroutine can yield,  
and @N{0 otherwise}.

```
}
```

```
@APIEntry{typedef @ldots lua_KContext;|
```

The type for continuation-function contexts.

It must be a numeric type.

This type is defined as @id{intptr\_t}

when @id{intptr\_t} is available,

so that it can store pointers too.

Otherwise, it is defined as @id{ptrdiff\_t}.

```
}
```

```
@APIEntry{
typedef int (*lua_KFunction) (lua_State *L, int status, lua_KContext ctx);|
```

Type for continuation functions @see{continuations}.

```
}
```

```
@APIEntry{void lua_len (lua_State *L, int index);|
@apii{0,1,e}}
```

Returns the length of the value at the given index.

It is equivalent to the @Char{#} operator in Lua @see{len-op} and  
may trigger a metamethod for the @Q{length} event @see{metatable}.

The result is pushed on the stack.

```
}
```

```
@APIEntry{
int lua_load (lua_State *L,
              lua_Reader reader,
```

```

        void *data,
        const char *chunkname,
        const char *mode);|
@apii{0,1,-}

```

Loads a Lua chunk without running it.

If there are no errors,

@id{lua\_load} pushes the compiled chunk as a Lua function on top of the stack.

Otherwise, it pushes an error message.

The @id{lua\_load} function uses a user-supplied @id{reader} function to read the chunk @seeC{lua\_Reader}.

The @id{data} argument is an opaque value passed to the reader function.

The @id{chunkname} argument gives a name to the chunk, which is used for error messages and in debug information @see{debugI}.

@id{lua\_load} automatically detects whether the chunk is text or binary and loads it accordingly (see program @idx{luac}).

The string @id{mode} works as in function @Lid{load}, with the addition that a @id{NULL} value is equivalent to the string @St{bt}.

@id{lua\_load} uses the stack internally, so the reader function must always leave the stack unmodified when returning.

@id{lua\_load} can return @Lid{LUA\_OK}, @Lid{LUA\_ERRSYNTAX}, or @Lid{LUA\_ERRMEM}. The function may also return other values corresponding to errors raised by the read function @see{statuscodes}.

If the resulting function has upvalues, its first upvalue is set to the value of the @x{global environment} stored at index @id{LUA\_RIDX\_GLOBALS} in the registry @see{registry}. When loading main chunks, this upvalue will be the @id{\_ENV} variable @see{globalenv}. Other upvalues are initialized with @nil.

```

}

```

```

@APIEntry{lua_State *lua_newstate (lua_Alloc f, void *ud);|
@apii{0,0,-}

```

Creates a new independent state and returns its main thread.

Returns @id{NULL} if it cannot create the state

(due to lack of memory).

The argument `@id{f}` is the `@x{allocator function}`;

Lua will do all memory allocation for this state

through this function `@seeF{lua_Alloc}`.

The second argument, `@id{ud}`, is an opaque pointer that Lua passes to the allocator in every call.

}

`@APIEntry{void lua_newtable (lua_State *L);|`

`@apii{0,1,m}`

Creates

a new empty table and pushes it onto the stack.

It is equivalent to `@T{lua_createtable(L, 0, 0)}`.

}

`@APIEntry{lua_State *lua_newthread (lua_State *L);|`

`@apii{0,1,m}`

Creates a new thread, pushes it on the stack,

and returns a pointer to a `@Lid{lua_State}` that represents this new thread.

The new thread returned by this function shares with the original thread

its global environment,

but has an independent execution stack.

Threads are subject to garbage collection,

like any Lua object.

}

`@APIEntry{void *lua_newuserdatauv (lua_State *L, size_t size, int nuvalue);|`

`@apii{0,1,m}`

This function creates and pushes on the stack a new full userdata,

with `@id{nuvalue}` associated Lua values, called `@id{user values}`,

plus an associated block of raw memory with `@id{size}` bytes.

(The user values can be set and read with the functions

`@Lid{lua_setiuservalue}` and `@Lid{lua_getiuservalue}`.)

The function returns the address of the block of memory.

Lua ensures that this address is valid as long as

the

corresponding userdata is alive `@see{GC}`.

Moreover, if the userdata is marked for finalization `@see{finalizers}`,

its address is valid at least until the call to its finalizer.

```
}
```

```
@APIEntry{int lua_next (lua_State *L, int index);|  
@apii{1,2|0,v}
```

Pops a key from the stack,  
and pushes a key@En{}value pair from the table at the given index,  
the @Q{next} pair after the given key.  
If there are no more elements in the table,  
then @Lid{lua\_next} @N{returns 0} and pushes nothing.

A typical table traversal looks like this:

```
@verbatim{  
/* table is in the stack at index 't' */  
lua_pushnil(L); /* first key */  
while (lua_next(L, t) != 0) {  
/* uses 'key' (at index -2) and 'value' (at index -1) */  
printf("%s - %s\n",  
    lua_typename(L, lua_type(L, -2)),  
    lua_typename(L, lua_type(L, -1)));  
/* removes 'value'; keeps 'key' for next iteration */  
lua_pop(L, 1);  
}  
}
```

While traversing a table,  
avoid calling @Lid{lua\_tolstring} directly on a key,  
unless you know that the key is  
actually a string.  
Recall that @Lid{lua\_tolstring} may change  
the value at the given index;  
this confuses the next call to @Lid{lua\_next}.

This function may raise an error if the given key  
is neither @nil nor present in the table.  
See function @Lid{next} for the caveats of modifying  
the table during its traversal.

```
}
```

```
@APIEntry{typedef @ldots lua_Number;|
```

The type of floats in Lua.

By default this type is double,  
but that can be changed to a single float or a long double.  
(See @id{LUA\_FLOAT\_TYPE} in @id{luaconf.h}.)

}

```
@APIEntry{int lua_numbertointeger (lua_Number n, lua_Integer *p);|
```

Tries to convert a Lua float to a Lua integer;  
the float `@id{n}` must have an integral value.  
If that value is within the range of Lua integers,  
it is converted to an integer and assigned to `@T{*p}`.  
The macro results in a boolean indicating whether the  
conversion was successful.  
(Note that this range test can be tricky to do  
correctly without this macro, due to rounding.)

This macro may evaluate its  
arguments more than once.

}

```
@APIEntry{int lua_pcall (lua_State *L, int nargs, int nresults, int msggh);|  
@apii{nargs + 1,nresults|1,-}
```

Calls a function (or a callable object) in protected mode.

Both `@id{nargs}` and `@id{nresults}` have the same meaning as  
in `@Lid{lua_call}`.

If there are no errors during the call,  
`@Lid{lua_pcall}` behaves exactly like `@Lid{lua_call}`.  
However, if there is any error,  
`@Lid{lua_pcall}` catches it,  
pushes a single value on the stack (the error object),  
and returns an error code.

Like `@Lid{lua_call}`,  
`@Lid{lua_pcall}` always removes the function  
and its arguments from the stack.

If `@id{msggh}` is 0,  
then the error object returned on the stack  
is exactly the original error object.  
Otherwise, `@id{msggh}` is the stack index of a  
`@emph{message handler}`.

(This index cannot be a pseudo-index.)  
In case of runtime errors,  
this handler will be called with the error object  
and its return value will be the object  
returned on the stack by `@Lid{lua_pcall}`.

Typically, the message

handler is used to add more debug information to the error object, such as a stack traceback. Such information cannot be gathered after the return of `@Lid{lua_pcall}`, since by then the stack has unwound.

The `@Lid{lua_pcall}` function returns one of the following status codes: `@Lid{LUA_OK}`, `@Lid{LUA_ERRRUN}`, `@Lid{LUA_ERRMEM}`, or `@Lid{LUA_ERRERR}`.

}

```
@APIEntry{
int lua_pcallk (lua_State *L,
               int nargs,
               int nresults,
               int msgch,
               lua_KContext ctx,
               lua_KFunction k);|
@apii{nargs + 1,nresults|1,-}
```

This function behaves exactly like `@Lid{lua_pcall}`, except that it allows the called function to yield `@see{continuations}`.

}

```
@APIEntry{void lua_pop (lua_State *L, int n);|
@apii{n,0,e}
```

Pops `@id{n}` elements from the stack. It is implemented as a macro over `@Lid{lua_settop}`.

}

```
@APIEntry{void lua_pushboolean (lua_State *L, int b);|
@apii{0,1,-}
```

Pushes a boolean value with value `@id{b}` onto the stack.

}

```
@APIEntry{void
lua_pushcclosure (lua_State *L, lua_CFunction fn, int n);|
@apii{n,1,m}
```

Pushes a new `@N{C closure}` onto the stack. This function receives a pointer to a `@N{C function}` and pushes onto the stack a Lua value of type `@id{function}` that, when called, invokes the corresponding `@N{C function}`. The parameter `@id{n}` tells how many upvalues this function will have



@see{c-closure}.

Any function to be callable by Lua must follow the correct protocol to receive its parameters and return its results @seeC{lua\_CFunction}.

When a @N{C function} is created, it is possible to associate some values with it, the so called upvalues; these upvalues are then accessible to the function whenever it is called. This association is called a @x{@N{C closure}} @see{c-closure}. To create a @N{C closure}, first the initial values for its upvalues must be pushed onto the stack. (When there are multiple upvalues, the first value is pushed first.) Then @Lid{lua\_pushcclosure} is called to create and push the @N{C function} onto the stack, with the argument @id{n} telling how many values will be associated with the function. @Lid{lua\_pushcclosure} also pops these values from the stack.

The maximum value for @id{n} is 255.

When @id{n} is zero, this function creates a @def{light @N{C function}}, which is just a pointer to the @N{C function}. In that case, it never raises a memory error.

}

```
@APIEntry{void lua_pushcfunction (lua_State *L, lua_CFunction f);|
@apii{0,1,-}}
```

Pushes a @N{C function} onto the stack. This function is equivalent to @Lid{lua\_pushcclosure} with no upvalues.

}

```
@APIEntry{const char *lua_pushfstring (lua_State *L, const char *fmt, ...);|
@apii{0,1,v}}
```

Pushes onto the stack a formatted string and returns a pointer to this string @see{constchar}. It is similar to the @ANSI{sprintf}, but has two important differences. First, you do not have to allocate space for the result; the result is a Lua string and Lua takes care of memory allocation

(and deallocation, through garbage collection).

Second,

the conversion specifiers are quite restricted.

There are no flags, widths, or precisions.

The conversion specifiers can only be

@Char{%%} (inserts the character @Char{%}),

@Char{%s} (inserts a zero-terminated string, with no size restrictions),

@Char{%f} (inserts a @Lid{lua\_Number}),

@Char{%I} (inserts a @Lid{lua\_Integer}),

@Char{%p} (inserts a pointer),

@Char{%d} (inserts an @T{int}),

@Char{%c} (inserts an @T{int} as a one-byte character), and

@Char{%U} (inserts a @T{long int} as a @x{UTF-8} byte sequence).

This function may raise errors due to memory overflow or an invalid conversion specifier.

}

```
@APIEntry{void lua_pushglobaltable (lua_State *L);|
@apii{0,1,-}}
```

Pushes the @x{global environment} onto the stack.

}

```
@APIEntry{void lua_pushinteger (lua_State *L, lua_Integer n);|
@apii{0,1,-}}
```

Pushes an integer with value @id{n} onto the stack.

}

```
@APIEntry{void lua_pushlightuserdata (lua_State *L, void *p);|
@apii{0,1,-}}
```

Pushes a light userdata onto the stack.

Userdata

represent @N{C values} in Lua.

A @def{light userdata} represents a pointer, a @T{void\*}.

It is a value (like a number):

you do not create it, it has no individual metatable,

and it is not collected (as it was never created).

A light userdata is equal to @Q{any}

light userdata with the same @N{C address}.

}

```
@APIEntry{const char *lua_pushliteral (lua_State *L, const char *s);|
@apii{0,1,m}
```

This macro is equivalent to @Lid{lua\_pushstring},  
but should be used only when @id{s} is a literal string.  
(Lua may optimize this case.)

}

```
@APIEntry{const char *lua_pushlstring (lua_State *L, const char *s, size_t len);|
@apii{0,1,m}
```

Pushes the string pointed to by @id{s} with size @id{len}  
onto the stack.  
Lua will make or reuse an internal copy of the given string,  
so the memory at @id{s} can be freed or reused immediately after  
the function returns.  
The string can contain any binary data,  
including @x{embedded zeros}.

Returns a pointer to the internal copy of the string @see{constchar}.

}

```
@APIEntry{void
lua_pushnil (lua_State *L);|
@apii{0,1,-}
```

Pushes a nil value onto the stack.

}

```
@APIEntry{void lua_pushnumber (lua_State *L, lua_Number n);|
@apii{0,1,-}
```

Pushes a float with value @id{n} onto the stack.

}

```
@APIEntry{const char *lua_pushstring (lua_State *L, const char *s);|
@apii{0,1,m}
```

Pushes the zero-terminated string pointed to by @id{s}  
onto the stack.  
Lua will make or reuse an internal copy of the given string,  
so the memory at @id{s} can be freed or reused immediately after

the function returns.

Returns a pointer to the internal copy of the string @see{constchar}.

If @id{s} is @id{NULL}, pushes @nil and returns @id{NULL}.

}

```
@APIEntry{int lua_pushthread (lua_State *L);|
@apii{0,1,-}}
```

Pushes the thread represented by @id{L} onto the stack.  
Returns 1 if this thread is the @x{main thread} of its state.

}

```
@APIEntry{void lua_pushvalue (lua_State *L, int index);|
@apii{0,1,-}}
```

Pushes a copy of the element at the given index  
onto the stack.

}

```
@APIEntry{
const
char *lua_pushvfstring (lua_State *L,
                        const char *fmt,
                        va_list argp);|
@apii{0,1,v}}
```

Equivalent to @Lid{lua\_pushfstring}, except that it receives a @id{va\_list}  
instead of a variable number of arguments.

}

```
@APIEntry{int lua_rawequal (lua_State *L, int index1, int index2);|
@apii{0,0,-}}
```

Returns 1 if the two values in indices @id{index1} and  
@id{index2} are primitively equal  
(that is, equal without calling the @idx{\_\_eq} metamethod).  
Otherwise @N{returns 0}.  
Also @N{returns 0} if any of the indices are not valid.

}

```
@APIEntry{int lua_rawget (lua_State *L, int index);|
```

@apii{1,1,-}

Similar to @Lid{lua\_gettable}, but does a raw access  
(i.e., without metamethods).

The value at @id{index} must be a table.

}

@APIEntry{int lua\_rawgeti (lua\_State \*L, int index, lua\_Integer n);|  
@apii{0,1,-}

Pushes onto the stack the value @T{t[n]},  
where @id{t} is the table at the given index.

The access is raw,  
that is, it does not use  
the @idx{\_\_index} metavalue.

Returns the type of the pushed value.

}

@APIEntry{int lua\_rawgetp (lua\_State \*L, int index, const void \*p);|  
@apii{0,1,-}

Pushes onto the stack the value @T{t[k]},  
where @id{t} is the table at the given index and  
@id{k} is the pointer @id{p} represented as a light userdata.  
The access is raw;  
that is, it does not use the @idx{\_\_index} metavalue.

Returns the type of the pushed value.

}

@APIEntry{lua\_Unsigned lua\_rawlen (lua\_State \*L, int index);|  
@apii{0,0,-}

Returns the raw @Q{length} of the value at the given index:  
for strings, this is the string length;  
for tables, this is the result of the length operator (@Char{#})  
with no metamethods;  
for userdata, this is the size of the block of memory allocated  
for the userdata.  
For other values, this call @N{returns 0}.

}

@APIEntry{void lua\_rawset (lua\_State \*L, int index);|

@apii{2,0,m}

Similar to @Lid{lua\_settable}, but does a raw assignment (i.e., without metamethods).

The value at @id{index} must be a table.

}

@APIEntry{void lua\_rawseti (lua\_State \*L, int index, lua\_Integer i);|  
@apii{1,0,m}

Does the equivalent of @T{t[i] = v},  
where @id{t} is the table at the given index  
and @id{v} is the value on the top of the stack.

This function pops the value from the stack.  
The assignment is raw,  
that is, it does not use the @idx{\_\_newindex} metavalue.

}

@APIEntry{void lua\_rawsetp (lua\_State \*L, int index, const void \*p);|  
@apii{1,0,m}

Does the equivalent of @T{t[p] = v},  
where @id{t} is the table at the given index,  
@id{p} is encoded as a light userdata,  
and @id{v} is the value on the top of the stack.

This function pops the value from the stack.  
The assignment is raw,  
that is, it does not use the @idx{\_\_newindex} metavalue.

}

@APIEntry{  
typedef const char \* (\*lua\_Reader) (lua\_State \*L,  
void \*data,  
size\_t \*size);|

The reader function used by @Lid{lua\_load}.  
Every time @Lid{lua\_load} needs another piece of  
the chunk,  
it calls the reader,  
passing along its @id{data} parameter.  
The reader must return a pointer to a block of memory  
with a new piece of the chunk

and set @id{size} to the block size.

The block must exist until the reader function is called again.

To signal the end of the chunk,

the reader must return @id{NULL} or set @id{size} to zero.

The reader function may return pieces of any size greater than zero.

```
}
```

```
@APIEntry{void lua_register (lua_State *L, const char *name, lua_CFunction f);|
@apii{0,0,e}
```

Sets the @N{C function} @id{f} as the new value of global @id{name}.

It is defined as a macro:

```
@verbatim{
#define lua_register(L,n,f) \
    (lua_pushcfunction(L, f), lua_setglobal(L, n))
}

}
```

```
@APIEntry{void lua_remove (lua_State *L, int index);|
@apii{1,0,-}
```

Removes the element at the given valid index,

shifting down the elements above this index to fill the gap.

This function cannot be called with a pseudo-index,

because a pseudo-index is not an actual stack position.

```
}
```

```
@APIEntry{void
lua_replace (lua_State *L, int index);|
@apii{1,0,-}
```

Moves the top element into the given valid index

without shifting any element

(therefore replacing the value at that given index),

and then pops the top element.

```
}
```

```
@APIEntry{int lua_resetthread (lua_State *L);|
@apii{0,?, -}
```

This function is deprecated;

it is equivalent to @Lid{lua\_closethread} with

@id{from} being @id{NULL}.

}

```
@APIEntry{int lua_resume (lua_State *L, lua_State *from, int nargs,  
                        int *nresults);|  
@apii{?,?,-}
```

Starts and resumes a coroutine in the given thread @id{L}.

To start a coroutine,  
you push the main function plus any arguments  
onto the empty stack of the thread.  
then you call @Lid{lua\_resume},  
with @id{nargs} being the number of arguments.  
This call returns when the coroutine suspends or finishes its execution.  
When it returns,  
@id{\*nresults} is updated and  
the top of the stack contains  
the @id{\*nresults} values passed to @Lid{lua\_yield}  
or returned by the body function.  
@Lid{lua\_resume}  
returns  
@Lid{LUA\_YIELD} if the coroutine yields,  
@Lid{LUA\_OK} if the coroutine finishes its execution  
without errors,  
or an error code in case of errors @see{statuscodes}.  
In case of errors,  
the error object is on the top of the stack.

To resume a coroutine,  
you remove the @id{\*nresults} yielded values from its stack,  
push the values to be passed as results from @id{yield},  
and then call @Lid{lua\_resume}.

The parameter @id{from} represents the coroutine that is resuming @id{L}.  
If there is no such coroutine,  
this parameter can be @id{NULL}.

}

```
@APIEntry{void lua_rotate (lua_State *L, int idx, int n);|  
@apii{0,0,-}
```

Rotates the stack elements between the valid index @id{idx}  
and the top of the stack.  
The elements are rotated @id{n} positions in the direction of the top,  
for a positive @id{n},  
or @T{-n} positions in the direction of the bottom,  
for a negative @id{n}.



The absolute value of @id{n} must not be greater than the size of the slice being rotated.

This function

cannot be called with a pseudo-index,  
because a pseudo-index is not an actual stack position.

}

```
@APIEntry{void lua_setallocf (lua_State *L, lua_Alloc f, void *ud);|
@apii{0,0,-}
```

Changes the @x{allocator function} of a given state to @id{f}  
with user data @id{ud}.

}

```
@APIEntry{void lua_setfield (lua_State *L, int index, const char *k);|
@apii{1,0,e}
```

Does the equivalent to @T{t[k] = v},  
where @id{t} is the value at the given index  
and @id{v} is the value on the top of the stack.

This function pops the value from the stack.  
As in Lua, this function may trigger a metamethod  
for the @Q{newindex} event @see{metatable}.

}

```
@APIEntry{void lua_setglobal (lua_State *L, const char *name);|
@apii{1,0,e}
```

Pops a value from the stack and  
sets it as the new value of global @id{name}.

}

```
@APIEntry{void lua_seti (lua_State *L, int index, lua_Integer n);|
@apii{1,0,e}
```

Does the equivalent to @T{t[n] = v},  
where @id{t} is the value at the given index  
and @id{v} is the value on the top of  
the stack.

This function pops the value from the stack.  
As in Lua, this function may trigger a metamethod  
for the @Q{newindex} event @see{metatable}.

}

```
@APIEntry{int lua_setiuservalue (lua_State *L, int index, int n);|
@apii{1,0,-}}
```

Pops a value from the stack and sets it as the new @id{n}-th user value associated to the full userdata at the given index. Returns 0 if the userdata does not have that value.

}

```
@APIEntry{int lua_setmetatable (lua_State *L, int index);|
@apii{1,0,-}}
```

Pops a table or @nil from the stack and sets that value as the new metatable for the value at the given index. (@nil means no metatable.)

(For historical reasons, this function returns an @id{int}, which now is always 1.)

}

```
@APIEntry{void lua_settable (lua_State *L, int index);|
@apii{2,0,e}}
```

Does the equivalent to @T{t[k] = v}, where @id{t} is the value at the given index, @id{v} is the value on the top of the stack, and @id{k} is the value just below the top.

This function pops both the key and the value from the stack. As in Lua, this function may trigger a metamethod for the @Q{newindex} event @see{metatable}.

}

```
@APIEntry{void lua_settop (lua_State *L, int index);|
@apii{?,?,e}}
```

Accepts any index, @N{or 0}, and sets the stack top to this index. If the new top is greater than the old one, then the new elements are filled with @nil. If @id{index} @N{is 0}, then all stack elements are removed.

This function can run arbitrary code when removing an index marked as to-be-closed from the stack.

```
}
```

```
@APIEntry{void lua_setwarnf (lua_State *L, lua_WarnFunction f, void *ud);|
@apii{0,0,-}
```

Sets the @x{warning function} to be used by Lua to emit warnings @see{lua\_WarnFunction}.  
The @id{ud} parameter sets the value @id{ud} passed to the warning function.

```
}
```

```
@APIEntry{typedef struct lua_State lua_State;|
```

An opaque structure that points to a thread and indirectly (through the thread) to the whole state of a Lua interpreter.  
The Lua library is fully reentrant:  
it has no global variables.  
All  
information about a state is accessible through this structure.

A pointer to this structure must be passed as the first argument to every function in the library, except to @Lid{lua\_newstate}, which creates a Lua state from scratch.

```
}
```

```
@APIEntry{int lua_status (lua_State *L);|
@apii{0,0,-}
```

Returns the status of the thread @id{L}.

The status can be @Lid{LUA\_OK} for a normal thread,  
an error code if the thread finished the execution  
of a @Lid{lua\_resume} with an error,  
or @Lid{LUA\_YIELD} if the thread is suspended.

You can call functions only in threads with status @Lid{LUA\_OK}.  
You can resume threads with status @Lid{LUA\_OK}  
(to start a new coroutine) or @Lid{LUA\_YIELD}  
(to resume a coroutine).

```
}
```

```
@APIEntry{size_t lua_stringtonumber (lua_State *L, const char *s);|
@apii{0,1,-}}
```

Converts the zero-terminated string @id{s} to a number,  
pushes that number into the stack,  
and returns the total size of the string,  
that is, its length plus one.  
The conversion can result in an integer  
or a float,  
according to the lexical conventions of Lua @see{lexical}.  
The string may have leading and trailing whitespaces and a sign.  
If the string is not a valid numeral,  
returns 0 and pushes nothing.  
(Note that the result can be used as a boolean,  
true if the conversion succeeds.)

```
}
```

```
@APIEntry{int lua_toboolean (lua_State *L, int index);|
@apii{0,0,-}}
```

Converts the Lua value at the given index to a @N{C boolean}  
value (@N{0 or 1}).  
Like all tests in Lua,  
@Lid{lua\_toboolean} returns true for any Lua value  
different from @false and @nil;  
otherwise it returns false.  
(If you want to accept only actual boolean values,  
use @Lid{lua\_isboolean} to test the value's type.)

```
}
```

```
@APIEntry{lua_CFunction lua_tocfunction (lua_State *L, int index);|
@apii{0,0,-}}
```

Converts a value at the given index to a @N{C function}.  
That value must be a @N{C function};  
otherwise, returns @id{NULL}.

```
}
```

```
@APIEntry{void lua_toclose (lua_State *L, int index);|
@apii{0,0,v}}
```

Marks the given index in the stack as a  
to-be-closed  
slot @see{to-be-closed}.  
Like a to-be-closed variable in Lua,

the value at that slot in the stack will be closed  
when it goes out of scope.  
Here, in the context of a C function,  
to go out of scope means that the running function returns to Lua,  
or there is an error,  
or the slot is removed from the stack through  
@Lid{lua\_settop} or @Lid{lua\_pop},  
or there is a call to @Lid{lua\_closeslot}.  
A slot marked as to-be-closed should not be removed from the stack  
by any other function in the API except @Lid{lua\_settop} or @Lid{lua\_pop},  
unless previously deactivated by @Lid{lua\_closeslot}.

This function raises an error if the value at the given slot  
neither has a @idx{\_\_close} metamethod nor is a false value.

This function should not be called for an index  
that is equal to or below an active to-be-closed slot.

Note that, both in case of errors and of a regular return,  
by the time the @idx{\_\_close} metamethod runs,  
the @N{C stack} was already unwound,  
so that any automatic @N{C variable}  
declared in the calling function  
(e.g., a buffer) will be out of scope.

}

@APIEntry{lua\_Integer lua\_tointeger (lua\_State \*L, int index);|  
@apii{0,0,-}}

Equivalent to @Lid{lua\_tointegerx} with @id{isnum} equal to @id{NULL}.

}

@APIEntry{lua\_Integer lua\_tointegerx (lua\_State \*L, int index, int \*isnum);|  
@apii{0,0,-}}

Converts the Lua value at the given index  
to the signed integral type @Lid{lua\_Integer}.  
The Lua value must be an integer,  
or a number or string convertible to an integer @see{coercion};  
otherwise, @id{lua\_tointegerx} @N{returns 0}.

If @id{isnum} is not @id{NULL},  
its referent is assigned a boolean value that  
indicates whether the operation succeeded.

}

```
@APIEntry{const char *lua_tolstring (lua_State *L, int index, size_t *len);|
@apii{0,0,m}
```

Converts the Lua value at the given index to a @N{C string}.

If @id{len} is not @id{NULL},

it sets @T{\*len} with the string length.

The Lua value must be a string or a number;

otherwise, the function returns @id{NULL}.

If

the value is a number,

then @id{lua\_tolstring} also

@emph{changes the actual value in the stack to a string}.

(This change confuses @Lid{lua\_next}

when @id{lua\_tolstring} is applied to keys during a table traversal.)

@id{lua\_tolstring} returns a pointer

to a string inside the Lua state @see{constchar}.

This string always has a zero (@Char{\0})

after its last character (as @N{in C}),

but can contain other zeros in its body.

This function can raise memory errors only

when converting a number to a string

(as then it may create a new string).

}

```
@APIEntry{lua_Number lua_tonumber (lua_State *L, int index);|
```

```
@apii{0,0,-}
```

Equivalent to @Lid{lua\_tonumberx} with @id{isnum} equal to @id{NULL}.

}

```
@APIEntry{lua_Number lua_tonumberx (lua_State *L, int index, int *isnum);|
```

```
@apii{0,0,-}
```

Converts the Lua value at the given index

to the @N{C type} @Lid{lua\_Number} @seeC{lua\_Number}.

The Lua value must be a number or a string convertible to a number

@see{coercion};

otherwise, @Lid{lua\_tonumberx}

@N{returns 0}.

If @id{isnum} is not @id{NULL},

its referent is assigned a boolean value that

indicates whether the operation succeeded.

```
}
```

```
@APIEntry{const void *lua_topointer (lua_State *L, int index);|  
@apii{0,0,-}}
```

Converts the value at the given index to a generic

@N{C pointer} (@T{void\*}).

The value can be a userdata, a table, a thread, a string, or a function;

otherwise, @id{lua\_topointer} returns @id{NULL}.

Different objects will give different pointers.

There is no way to convert the pointer back to its original value.

Typically this function is used only for hashing and debug information.

```
}
```

```
@APIEntry{const char *lua_tostring (lua_State *L, int index);|  
@apii{0,0,m}}
```

Equivalent to @Lid{lua\_tolstring} with @id{len} equal to @id{NULL}.

```
}
```

```
@APIEntry{lua_State *lua_tothread (lua_State *L, int index);|  
@apii{0,0,-}}
```

Converts the value at the given index to a Lua thread

(represented as @T{lua\_State\*}).

This value must be a thread;

otherwise, the function returns @id{NULL}.

```
}
```

```
@APIEntry{void  
*lua_touserdata (lua_State *L, int index);|  
@apii{0,0,-}}
```

If the value at the given index is a full userdata,  
returns its memory-block address.

If the value is a light userdata,  
returns its value (a pointer).

Otherwise, returns @id{NULL}.

```
}
```

```
@APIEntry{int lua_type (lua_State *L, int index);|  
@apii{0,0,-}}
```

Returns the type of the value in the given valid index,  
or @id{LUA\_TNONE} for a non-valid but acceptable index.  
The types returned by @Lid{lua\_type} are coded by the following constants  
defined in @id{lua.h}:

```
@defid{LUA_TNIL},
@defid{LUA_TNUMBER},
@defid{LUA_TBOOLEAN},
@defid{LUA_TSTRING},
@defid{LUA_TTABLE},
@defid{LUA_TFUNCTION},
@defid{LUA_TUSERDATA},
@defid{LUA_TTHREAD},
and
@defid{LUA_TLIGHTUSERDATA}.
```

```
}
```

```
@APIEntry{const char *lua_typename (lua_State *L, int tp);|
@apii{0,0,-}}
```

Returns the name of the type encoded by the value @id{tp},  
which must be one the values returned by @Lid{lua\_type}.

```
}
```

```
@APIEntry{typedef @ldots lua_Unsigned;|
```

The unsigned version of @Lid{lua\_Integer}.

```
}
```

```
@APIEntry{int
lua_upvalueindex (int i);|
@apii{0,0,-}}
```

Returns the pseudo-index that represents the @id{i}-th upvalue of  
the running function @see{c-closure}.  
@id{i} must be in the range @M{[1,256]}.

```
}
```

```
@APIEntry{lua_Number lua_version (lua_State *L);|
@apii{0,0,-}}
```

Returns the version number of this core.

```
}
```



```
@APIEntry{
typedef void (*lua_WarnFunction) (void *ud, const char *msg, int tocont);|
```

The type of @x{warning function}s, called by Lua to emit warnings.

The first parameter is an opaque pointer  
set by @Lid{lua\_setwarnf}.

The second parameter is the warning message.

The third parameter is a boolean that  
indicates whether the message is  
to be continued by the message in the next call.

See @Lid{warn} for more details about warnings.

```
}
```

```
@APIEntry{
void lua_warning (lua_State *L, const char *msg, int tocont);|
@apii{0,0,-}
```

Emits a warning with the given message.

A message in a call with @id{tocont} true should be  
continued in another call to this function.

See

@Lid{warn} for more details about warnings.

```
}
```

```
@APIEntry{
typedef int (*lua_Writer) (lua_State *L,
                           const void* p,
                           size_t sz,
                           void* ud);|
```

The type of the writer function used by @Lid{lua\_dump}.

Every time @Lid{lua\_dump} produces another piece of chunk,  
it calls the writer,  
passing along the buffer to be written (@id{p}),  
its size (@id{sz}),  
and the @id{ud} parameter supplied to @Lid{lua\_dump}.

The writer returns an error code:

@N{0 means} no errors;  
any other value means an error and stops @Lid{lua\_dump} from  
calling the writer again.

```
}
```

```
@APIEntry{void lua_xmove (lua_State *from, lua_State *to, int n);|
@apii{?,?,-}}
```

Exchange values between different threads of the same state.

This function pops @id{n} values from the stack @id{from},  
and pushes them onto the stack @id{to}.

```
}
```

```
@APIEntry{int lua_yield (lua_State *L, int nresults);|
@apii{?,?,v}}
```

This function is equivalent to @Lid{lua\_yieldk},  
but it has  
no continuation @see{continuations}.  
Therefore, when the thread resumes,  
it continues the function that called  
the function calling @id{lua\_yield}.  
To avoid surprises,  
this function should be called only in a tail call.

```
}
```

```
@APIEntry{
int lua_yieldk (lua_State *L,
               int nresults,
               lua_KContext ctx,
               lua_KFunction k);|
@apii{?,?,v}}
```

Yields a coroutine (thread).

When a @N{C function} calls @Lid{lua\_yieldk},  
the running coroutine suspends its execution,  
and the call to @Lid{lua\_resume} that started this coroutine returns.  
The parameter @id{nresults} is the number of values from the stack  
that will be passed as results to @Lid{lua\_resume}.

When the coroutine is resumed again,  
Lua calls the given @x{continuation function} @id{k} to continue  
the execution of the @N{C function} that yielded @see{continuations}.  
This continuation function receives the same stack  
from the previous function,  
with the @id{n} results removed and  
replaced by the arguments

passed to @Lid{lua\_resume}.

Moreover,

the continuation function receives the value @id{ctx}  
that was passed to @Lid{lua\_yieldk}.

Usually, this function does not return;  
when the coroutine eventually resumes,  
it continues executing the continuation function.

However, there is one special case,  
which is when this function is called  
from inside a line or a count hook @see{debugI}.

In that case, @id{lua\_yieldk} should be called with no continuation  
(probably in the form of @Lid{lua\_yield}) and no results,  
and the hook should return immediately after the call.

Lua will yield and,  
when the coroutine resumes again,  
it will continue the normal execution  
of the (Lua) function that triggered the hook.

This function can raise an error if it is called from a thread  
with a pending C call with no continuation function  
(what is called a @emphx{C-call boundary}),  
or it is called from a thread that is not running inside a resume  
(typically the main thread).

}

}

@sect2{debugI| @title{The Debug  
Interface}}

Lua has no built-in debugging facilities.  
Instead, it offers a special interface  
by means of functions and @emph{hooks}.  
This interface allows the construction of different  
kinds of debuggers, profilers, and other tools  
that need @Q{inside information} from the interpreter.

```
@APIEntry{
typedef struct lua_Debug {
    int event;
    const char *name;      /* (n) */
    const char *namewhat;  /* (n) */
    const char *what;      /* (S) */
    const char *source;    /* (S) */
    size_t srclen;         /* (S) */
}
```

```

int currentline;      /* (l) */
int linedefined;      /* (S) */
int lastlinedefined;  /* (S) */
unsigned char nups;    /* (u) number of upvalues */
unsigned char nparams; /* (u) number of parameters */
char isvararg;         /* (u) */
char istailcall;       /* (t) */
unsigned short ftransfer; /* (r) index of first value transferred */
unsigned short ntransfer; /* (r) number of transferred values
*/
char short_src[LUA_IDSIZE]; /* (S) */
/* private part */
@rep{other fields}
} lua_Debug;
|

```

A structure used to carry different pieces of information about a function or an activation record.

@Lid{lua\_getstack} fills only the private part of this structure, for later use.

To fill the other fields of @Lid{lua\_Debug} with useful information, you must call @Lid{lua\_getinfo} with an appropriate parameter.

(Specifically, to get a field, you must add the letter between parentheses in the field's comment to the parameter @id{what} of @Lid{lua\_getinfo}.)

The fields of @Lid{lua\_Debug} have the following meaning:

@description{

@item{@id{source}|

the source of the chunk that created the function.

If @T{source} starts with a @Char{@At},

it means that the function was defined in a file where

the file name follows the @Char{@At}.

If @T{source} starts with a @Char{=},

the remainder of its contents describes the source in a user-dependent manner.

Otherwise,

the function was defined in a string

where

@T{source} is that string.

}

@item{@id{srclen}|

The length of the string @id{source}.

}

@item{@id{short\_src}|

a @Q{printable} version of @T{source}, to be used in error messages.

}

@item{@id{linedefined}|

the line number where the definition of the function starts.

}

@item{@id{lastlinedefined}|

the line number where the definition of the function ends.

}

@item{@id{what}|

the string @T{"Lua"} if the function is a Lua function,

@T{"C"} if it is a @N{C function},

@T{"main"} if it is the main part of a chunk.

}

@item{@id{currentline}|

the current line where the given function is executing.

When no line information is available,

@T{currentline} is set to @num{-1}.

}

@item{@id{name}|

a reasonable name for the given function.

Because functions in Lua are first-class values,

they do not have a fixed name:

some functions can be the value of multiple global variables,

while others can be stored only in a table field.

The @T{lua\_getinfo} function checks how the function was called

to find a suitable name.

If it cannot find a name,

then @id{name} is set to @id{NULL}.

}

@item{@id{namewhat}|

explains the @T{name} field.

The value of @T{namewhat} can be

@T{"global"}, @T{"local"}, @T{"method"},

@T{"field"}, @T{"upvalue"}, or @T{""} (the empty string),

according to how the function was called.

(Lua uses the empty string when no other option seems to apply.)

}

@item{@id{istailcall}|

true if this function invocation was called by a tail call.

In this case, the caller of this level is not in the stack.

}

@item{@id{nups}|  
the number of upvalues of the function.  
}

@item{@id{nparams}|  
the number of parameters of the function  
(always @N{0 for} @N{C functions}).  
}

@item{@id{isvararg}|  
true if the function is a variadic function  
(always true for @N{C functions}).  
}

@item{@id{ftransfer}|  
the index in the stack of the first value being @Q{transferred},  
that is, parameters in a call or return values in a return.  
(The other values are in consecutive indices.)  
Using this  
index, you can access and modify these values  
through @Lid{lua\_getlocal} and @Lid{lua\_setlocal}.  
This field is only meaningful during a  
call hook, denoting the first parameter,  
or a return hook, denoting the first value being returned.  
(For call hooks, this value is always 1.)  
}

@item{@id{ntransfer}|  
The number of values being transferred (see previous item).  
(For calls of Lua functions,  
this value is always equal to @id{nparams}.)  
}

}

}

@APIEntry{lua\_Hook lua\_gethook (lua\_State \*L);|  
@apii{0,0,-}}

Returns the current hook function.

}

@APIEntry{int lua\_gethookcount (lua\_State \*L);|  
@apii{0,0,-}}

Returns the current hook count.

```
}
```

```
@APIEntry{int lua_gethookmask (lua_State *L);|
@apii{0,0,-}
```

Returns the current hook mask.

```
}
```

```
@APIEntry{int lua_getinfo (lua_State *L, const char *what, lua_Debug *ar);|
@apii{0|1,0|1|2,m}
```

Gets information about a specific function or function invocation.

To get information about a function invocation,  
the parameter `@id{ar}` must be a valid activation  
record that was  
filled by a previous call to `@Lid{lua_getstack}` or  
given as argument to a hook `@seeC{lua_Hook}`.

To get information about a function, you push it onto the stack  
and start the `@id{what}` string with the character `@Char{>}`.  
(In that case,  
`@id{lua_getinfo}` pops the function from the top of the stack.)  
For instance, to know in which line a function `@id{f}` was defined,  
you can write the following code:

```
@verbatim{
lua_Debug ar;
lua_getglobal(L, "f"); /* get global 'f' */
lua_getinfo(L, ">S", &ar);
printf("%d\n", ar.linedefined);
}
```

Each character in the string `@id{what}`  
selects some fields of the structure `@id{ar}` to be filled or  
a value to be pushed on the stack.  
(These characters are also documented in the declaration of  
the structure `@Lid{lua_Debug}`,  
between parentheses in the comments following each field.)  
`@description{`

```
@item{@Char{f}|
pushes onto the stack the function that is
running at the given level;
}
```

```
@item{@Char{1}| fills in the field @id{currentline};  
}
```

```
@item{@Char{n}|  
fills in the fields @id{name} and @id{namewhat};  
}
```

```
@item{@Char{r}| fills in the fields @id{ftransfer} and @id{ntransfer};  
}
```

```
@item{@Char{S}|  
fills in the fields @id{source}, @id{short_src},  
@id{linedefined}, @id{lastlinedefined}, and @id{what};  
}
```

```
@item{@Char{t}| fills in the field @id{istailcall};  
}
```

```
@item{@Char{u}| fills in the fields  
@id{nups}, @id{nparams}, and @id{isvararg};  
}
```

```
@item{@Char{L}|  
pushes onto the stack a table whose indices are  
the lines on the function with some associated code,  
that is, the lines where you can put a break point.  
(Lines with no code include empty lines and comments.)  
If this option is given together with option @Char{f},  
its table is pushed after the function.  
This is the only option that can raise a memory error.  
}
```

```
}
```

This function returns 0 to signal an invalid option in @id{what};  
even then the valid options are handled correctly.

```
}
```

```
@APIEntry{const char *lua_getlocal (lua_State *L, const lua_Debug *ar, int n);|  
@apii{0,0|1,-}}
```

Gets  
information about a local variable or a temporary value  
of a given activation record or a given function.

In the first case,  
the parameter @id{ar} must be a valid activation record that was



filled by a previous call to `@Lid{lua_getstack}` or given as argument to a hook `@seeC{lua_Hook}`. The index `@id{n}` selects which local variable to inspect; see `@Lid{debug.getlocal}` for details about variable indices and names.

`@Lid{lua_getlocal}` pushes the variable's value onto the stack and returns its name.

In the second case, `@id{ar}` must be `@id{NULL}` and the function to be inspected must be on the top of the stack.

In this case, only parameters of Lua functions are visible (as there is no information about what variables are active) and no values are pushed onto the stack.

Returns `@id{NULL}` (and pushes nothing) when the index is greater than the number of active local variables.

}

```
@APIEntry{int lua_getstack (lua_State *L, int level, lua_Debug *ar);|
@apii{0,0,-}
```

Gets information about the interpreter runtime stack.

This function fills parts of a `@Lid{lua_Debug}` structure with an identification of the `@emph{activation record}` of the function executing at a given level. `@N{Level 0}` is the current running function, whereas level `@M{n+1}` is the function that has called level `@M{n}` (except for tail calls, which do not count in the stack). When called with a level greater than the stack depth, `@Lid{lua_getstack}` returns 0; otherwise it returns 1.

}

```
@APIEntry{const char *lua_getupvalue (lua_State *L, int funcindex, int n);|
@apii{0,0|1,-}
```

Gets information about the `@id{n}`-th upvalue of the closure at index `@id{funcindex}`. It pushes the upvalue's value onto the stack and returns its name. Returns `@id{NULL}` (and pushes nothing) when the index `@id{n}` is greater than the number of upvalues.

See @Lid{debug.getupvalue} for more information about upvalues.

}

@APIEntry{typedef void (\*lua\_Hook) (lua\_State \*L, lua\_Debug \*ar);|

Type for debugging hook functions.

Whenever a hook

is called, its @id{ar} argument has its field

@id{event} set to the specific event that triggered the hook.

Lua identifies these events with the following constants:

@defid{LUA\_HOOKCALL}, @defid{LUA\_HOOKRET},

@defid{LUA\_HOOKTAILCALL}, @defid{LUA\_HOOKLINE},

and @defid{LUA\_HOOKCOUNT}.

Moreover, for line events, the field @id{currentline} is also set.

To get the value of any other field in @id{ar},

the hook must call @Lid{lua\_getinfo}.

For call events, @id{event} can be @id{LUA\_HOOKCALL},

the normal value, or @id{LUA\_HOOKTAILCALL}, for a tail call;

in this case, there will be no corresponding return event.

While Lua is running a hook, it disables other calls to hooks.

Therefore, if a hook calls back Lua to execute a function or a chunk,  
this execution occurs without any calls to hooks.

Hook functions cannot have continuations,

that is, they cannot call @Lid{lua\_yieldk},

@Lid{lua\_pcallk}, or @Lid{lua\_callk} with a non-null @id{k}.

Hook functions can yield under the following conditions:

Only

count and line events can yield;

to yield, a hook function must finish its execution

calling @Lid{lua\_yield} with @id{nresults} equal to zero

(that is, with no values).

}

@APIEntry{void lua\_sethook (lua\_State \*L, lua\_Hook f, int mask, int count);|

@apii{0,0,-}

Sets the debugging hook function.

Argument @id{f} is the hook function.

@id{mask} specifies on which events the hook will be called:

it is formed by a bitwise OR of the constants

@defid{LUA\_MASKCALL},

@defid{LUA\_MASKRET},

@defid{LUA\_MASKLINE},

and @defid{LUA\_MASKCOUNT}.

The @id{count} argument is only meaningful when the mask includes @id{LUA\_MASKCOUNT}.

For each event, the hook is called as explained below:

@description{

@item{The call hook| is called when the interpreter calls a function.

The hook is called just after Lua enters the new function.

}

@item{The return hook| is called when the interpreter returns from a function.

The hook is called just before Lua leaves the function.

}

@item{The line hook| is called

when the interpreter is about to

start the execution of a new line of code,

or when it jumps back in the code (even to the same line).

This event only happens while Lua is executing a Lua function.

}

@item{The count hook| is called after the interpreter executes every

@T{count} instructions.

This event only happens while Lua is executing a Lua function.

}

}

Hooks are disabled by setting @id{mask} to zero.

}

@APIEntry{const char \*lua\_setlocal (lua\_State \*L, const lua\_Debug \*ar, int n);|

@apii{0|1,0,-}}

Sets the value of a local variable of a given activation record.

It assigns the value on the top of the stack

to the variable and returns its name.

It also pops the value from the stack.

Returns @id{NULL} (and pops nothing)

when the index is greater than

the number of active local variables.

Parameters @id{ar} and @id{n} are as in the function @Lid{lua\_getlocal}.

}

```
@APIEntry{const char *lua_setupvalue (lua_State *L, int funcindex, int n);|
@apii{0|1,0,-}
```

Sets the value of a closure's  
upvalue.

It assigns the value on the top of the stack  
to the upvalue and returns its name.

It also pops the value from the stack.

Returns @id{NULL} (and pops nothing)  
when the index @id{n} is greater than the number of upvalues.

Parameters @id{funcindex} and @id{n} are as in  
the function @Lid{lua\_getupvalue}.

}

```
@APIEntry{void *lua_upvalueid (lua_State *L, int funcindex, int n);|
@apii{0,0,-}
```

Returns a unique identifier for the upvalue numbered @id{n}  
from the closure at index @id{funcindex}.

These unique identifiers allow a program to check whether different  
closures share upvalues.

Lua closures that share an upvalue  
(that is, that access a same external local variable)  
will return identical ids for those upvalue indices.

Parameters @id{funcindex} and @id{n} are as in  
the function @Lid{lua\_getupvalue},  
but @id{n} cannot be greater than the number of upvalues.

}

```
@APIEntry{
void lua_upvaluejoin (lua_State *L, int funcindex1, int n1,
                    int
                    funcindex2, int n2);|
@apii{0,0,-}
```

Make the @id{n1}-th upvalue of the Lua closure at index @id{funcindex1}  
refer to the @id{n2}-th upvalue of the Lua closure at index @id{funcindex2}.

}

}

}

@C{-----}

@sect1{auxlib|@title{The Auxiliary Library}}

@simplesect{

@index{luauxlib.h}

The @def{auxiliary library} provides several convenient functions to interface C with Lua.

While the basic API provides the primitive functions for all interactions between C and Lua, the auxiliary library provides higher-level functions for some common tasks.

All functions and types from the auxiliary library are defined in header file @id{luauxlib.h} and have a prefix @id{luaL\_}.

All functions in the auxiliary library are built on top of the basic API, and so they provide nothing that cannot be done with that API. Nevertheless, the use of the auxiliary library ensures more consistency to your code.

Several functions in the auxiliary library use internally some extra stack slots. When a function in the auxiliary library uses less than five slots, it does not check the stack size; it simply assumes that there are enough slots.

Several functions in the auxiliary library are used to check @N{C function} arguments. Because the error message is formatted for arguments (e.g., @St{bad argument #1}), you should not use these functions for other stack values.

Functions called @id{luaL\_check\*} always raise an error if the check is not satisfied.

}

@sect2{@title{Functions and Types}}

Here we list all functions and types from the auxiliary library in alphabetical order.

@APIEntry{void luaL\_addchar (luaL\_Buffer \*B, char c);|  
@apii{?,?,m}}

Adds the byte @id{c} to the buffer @id{B}  
@seeC{luaL\_Buffer}.

}

@APIEntry{  
const void luaL\_addgsub (luaL\_Buffer \*B, const char \*s,  
const char \*p, const char \*r);|  
@apii{?,?,m}}

Adds a copy of the string @id{s} to the buffer @id{B} @seeC{luaL\_Buffer},  
replacing any  
occurrence of the string @id{p}  
with the string @id{r}.

}

@APIEntry{void luaL\_addlstring (luaL\_Buffer \*B, const char \*s, size\_t l);|  
@apii{?,?,m}}

Adds the string pointed to by @id{s} with length @id{l} to  
the buffer @id{B}  
@seeC{luaL\_Buffer}.

The string can contain @x{embedded zeros}.

}

@APIEntry{void luaL\_addsize (luaL\_Buffer \*B, size\_t n);|  
@apii{?,?,-}}

Adds to the buffer @id{B}  
a string of length @id{n} previously copied to the  
buffer area @seeC{luaL\_prepbuffer}.

}

@APIEntry{void luaL\_addstring (luaL\_Buffer \*B, const char \*s);|

@apii{?,?,m}

Adds the zero-terminated string pointed to by @id{s}  
to the buffer @id{B}  
@seeC{luaL\_Buffer}.

}

@APIEntry{void luaL\_addvalue (luaL\_Buffer \*B);|  
@apii{?,?,m}

Adds the value on the top of the stack  
to the buffer @id{B}  
@seeC{luaL\_Buffer}.  
Pops the value.

This is the only function on string buffers that can (and must)  
be called with an extra element on the stack,  
which is the value to be added to the buffer.

}

@APIEntry{  
void luaL\_argcheck  
(lua\_State \*L,  
int cond,  
int arg,  
const char \*extramsg);|  
@apii{0,0,v}

Checks whether @id{cond} is true.  
If it is not, raises an error with a standard message @seeF{luaL\_argerror}.

}

@APIEntry{int luaL\_argerror (lua\_State \*L, int arg, const char \*extramsg);|  
@apii{0,0,v}

Raises an error reporting a problem with argument @id{arg}  
of the @N{C function} that called it,  
using a standard message  
that includes @id{extramsg} as a comment:  
@verbatim{  
bad argument #@rep{arg} to '@rep{funcname}' (@rep{extramsg})  
}  
This function never returns.

}

```
@APIEntry{
void luaL_argexpected (lua_State *L,
                        int cond,
                        int arg,
                        const char *tname);|
@apii{0,0,v}
```

Checks whether @id{cond} is true.

If it is not, raises an error about the type of the argument @id{arg} with a standard message @seeF{luaL\_typeerror}.

```
}
```

```
@APIEntry{typedef struct luaL_Buffer luaL_Buffer;|
```

Type for

a @def{string buffer}.

A string buffer allows @N{C code} to build Lua strings piecemeal.

Its pattern of use is as follows:

```
@itemize{
```

```
@item{First declare a variable @id{b} of type @Lid{luaL_Buffer}.}
```

```
@item{Then initialize it with a call @T{luaL_buffinit(L, &b)}.
```

```
@item{
```

Then add string pieces to the buffer calling any of the @id{luaL\_add\*} functions.

```
}
```

```
@item{
```

Finish by calling @T{luaL\_pushresult(&b)}.

This call leaves the final string on the top of the stack.

```
}
```

```
}
```

If you know beforehand the maximum size of the resulting string, you can use the buffer like this:

```
@itemize{
```

```
@item{First declare a variable @id{b} of type @Lid{luaL_Buffer}.}
```

```
@item{Then initialize it and preallocate a space of
size @id{sz} with a call @T{luaL_buffinitsize(L, &b, sz)}.
```



```
@item{Then produce the string into that space.}
```

```
@item{  
Finish by calling @T{luaL_pushresultsize(&b, sz)},  
where @id{sz} is the total size of the resulting string  
copied into that space (which may be less than or  
equal to the preallocated  
size).  
}  
  
}
```

During its normal operation,  
a string buffer uses a variable number of stack slots.  
So, while using a buffer, you cannot assume that you know where  
the top of the stack is.  
You can use the stack between successive calls to buffer operations  
as long as that use is balanced;  
that is,  
when you call a buffer operation,  
the stack is at the same level  
it was immediately after the previous buffer operation.  
(The only exception to this rule is @Lid{luaL\_addvalue}.)  
After calling @Lid{luaL\_pushresult},  
the stack is back to its level when the buffer was initialized,  
plus the final string on its top.

```
}
```

```
@APIEntry{char *luaL_buffaddr (luaL_Buffer *B);|  
@apii{0,0,-}}
```

Returns the address of the current content of buffer @id{B}  
@seeC{luaL\_Buffer}.  
Note that any addition to the buffer may invalidate this address.

```
}
```

```
@APIEntry{void luaL_buffinit (lua_State *L, luaL_Buffer *B);|  
@apii{0,?,-}}
```

Initializes a buffer @id{B}  
@seeC{luaL\_Buffer}.  
This function does not allocate any space;  
the  
buffer must be declared as a variable.

```
}
```

```
@APIEntry{size_t luaL_bufflen (luaL_Buffer *B);|
@apii{0,0,-}
```

Returns the length of the current content of buffer @id{B}  
@seeC{luaL\_Buffer}.

```
}
```

```
@APIEntry{char *luaL_buffinitsize (lua_State *L, luaL_Buffer *B, size_t sz);|
@apii{?,?,m}
```

Equivalent to the sequence  
@Lid{luaL\_buffinit}, @Lid{luaL\_prepbuffsize}.

```
}
```

```
@APIEntry{void luaL_buffsub (luaL_Buffer *B, int n);|
@apii{?,?,-}
```

Removes @id{n} bytes from the buffer @id{B}  
@seeC{luaL\_Buffer}.  
The buffer must have at least that many bytes.

```
}
```

```
@APIEntry{int luaL_callmeta (lua_State *L, int obj, const char *e);|
@apii{0,0|1,e}
```

Calls a metamethod.

If the object at index @id{obj} has a metatable and this  
metatable has a field @id{e},  
this function calls this field passing the object as its only argument.  
In this case this function returns true and pushes onto the  
stack the value returned by the call.  
If there is no metatable or no metamethod,  
this function returns false without pushing any value  
on the stack.

```
}
```

```
@APIEntry{void luaL_checkany (lua_State *L, int arg);|
@apii{0,0,v}
```

Checks whether the function has an argument  
of any type (including @nil) at position @id{arg}.

```
}
```

```
@APIEntry{lua_Integer luaL_checkinteger (lua_State *L, int arg);|
@apii{0,0,v}
```

Checks whether the function argument @id{arg} is an integer  
(or can be converted to an integer)  
and returns this integer.

```
}
```

```
@APIEntry{const char *luaL_checklstring (lua_State *L, int arg, size_t *l);|
@apii{0,0,v}
```

Checks whether the function argument @id{arg} is a string  
and returns this string;  
if @id{l} is not @id{NULL} fills its referent  
with the string's length.

This function uses @Lid{lua\_tolstring} to get its result,  
so all conversions and caveats of that function apply here.

```
}
```

```
@APIEntry{lua_Number luaL_checknumber (lua_State *L, int arg);|
@apii{0,0,v}
```

Checks whether the function argument @id{arg} is a number  
and returns this number converted to a @id{lua\_Number}.

```
}
```

```
@APIEntry{
int luaL_checkoption (lua_State
*L,
    int arg,
    const char *def,
    const char *const lst[]);|
@apii{0,0,v}
```

Checks whether the function argument @id{arg} is a string and  
searches for this string in the array @id{lst}  
(which must be NULL-terminated).  
Returns the index in the array where the string was found.  
Raises an error if the argument is not a string or  
if the string cannot be found.

If @id{def} is not @id{NULL},

the function uses `@id{def}` as a default value when there is no argument `@id{arg}` or when this argument is `@nil`.

This is a useful function for mapping strings to `@N{C enums}`.  
(The usual convention in Lua libraries is to use strings instead of numbers to select options.)

}

```
@APIEntry{void luaL_checkstack (lua_State *L, int sz, const char *msg);|
@apii{0,0,v}
```

Grows the stack size to `@T{top + sz}` elements, raising an error if the stack cannot grow to that size.  
`@id{msg}` is an additional text to go into the error message (or `@id{NULL}` for no additional text).

}

```
@APIEntry{const char *luaL_checkstring (lua_State *L, int arg);|
@apii{0,0,v}
```

Checks whether the function argument `@id{arg}` is a string and returns this string.

This function uses `@Lid{lua_tolstring}` to get its result, so all conversions and caveats of that function apply here.

}

```
@APIEntry{void luaL_checktype (lua_State *L, int arg, int t);|
@apii{0,0,v}
```

Checks whether the function argument `@id{arg}` has type `@id{t}`. See `@Lid{lua_type}` for the encoding of types for `@id{t}`.

}

```
@APIEntry{void *luaL_checkudata (lua_State *L, int arg, const char *tname);|
@apii{0,0,v}
```

Checks whether the function argument `@id{arg}` is a userdata of the type `@id{tname}` `@seeC{luaL_newmetatable}` and returns the userdata's memory-block address `@seeC{lua_touserdata}`.

}

```
@APIEntry{void luaL_checkversion (lua_State *L);|
@apii{0,0,v}
```

Checks whether the code making the call and the Lua library being called are using the same version of Lua and the same numeric types.

```
}
```

```
@APIEntry{int
luaL_dofile (lua_State *L, const char *filename);|
@apii{0,?,m}
```

Loads and runs the given file.

It is defined as the following macro:

```
@verbatim{
(luaL_loadfile(L, filename) || lua_pcall(L, 0, LUA_MULTRET, 0))
}
```

It @N{returns 0} (@Lid{LUA\_OK}) if there are no errors,  
or 1 in case of errors.

```
}
```

```
@APIEntry{int luaL_dostring (lua_State *L, const char *str);|
@apii{0,?,-}
```

Loads and runs the given string.

It is defined as the following macro:

```
@verbatim{
(luaL_loadstring(L, str) || lua_pcall(L, 0, LUA_MULTRET, 0))
}
```

It @N{returns 0} (@Lid{LUA\_OK}) if there are no errors,  
or 1 in case of errors.

```
}
```

```
@APIEntry{int luaL_error (lua_State *L, const char *fmt, ...);|
@apii{0,0,v}
```

Raises an error.

The error message format is given by @id{fmt}

plus any extra arguments,

following the same rules of @Lid{lua\_pushfstring}.

It also adds at the beginning of the message the file name and  
the line number where the error occurred,  
if this information is available.

This function never returns,

but it is an

idiom to use it in @N{C functions}  
as @T{return luaL\_error(@rep{args})}).

}

@APIEntry{int luaL\_execresult (lua\_State \*L, int stat);|  
@apii{0,3,m}}

This function produces the return values for  
process-related functions in the standard library  
(@Lid{os.execute} and @Lid{io.close}).

}

@APIEntry{  
int luaL\_fileresult (lua\_State \*L, int stat, const char \*fname);|  
@apii{0,1|3,m}}

This function produces the return values for  
file-related functions in the standard library  
(@Lid{io.open}, @Lid{os.rename}, @Lid{file:seek}, etc.).

}

@APIEntry{int luaL\_getmetafield (lua\_State \*L, int obj, const char \*e);|  
@apii{0,0|1,m}}

Pushes onto the stack the field @id{e} from the metatable  
of the object at index @id{obj} and returns the type of the pushed value.  
If the object does not have a metatable,  
or if the metatable does not have this field,  
pushes nothing and returns @id{LUA\_TNIL}.

}

@APIEntry{int luaL\_getmetatable (lua\_State \*L, const char \*tname);|  
@apii{0,1,m}}

Pushes onto the stack the metatable  
associated with the name @id{tname}  
in the registry @seeC{luaL\_newmetatable},  
or @nil if there is no metatable associated with that name.  
Returns the type of the pushed value.

}

@APIEntry{int luaL\_getsubtable (lua\_State \*L, int idx, const char \*fname);|  
@apii{0,1,e}}

Ensures that the value `@T{t[fname]}`,  
 where `@id{t}` is the value at index `@id{idx}`,  
 is a table,  
 and pushes that table onto the stack.  
 Returns true if it finds a previous table there  
 and false if it creates a new table.

}

@APIEntry{  
 const char \*luaL\_gsub (lua\_State \*L,  
     const char \*s,  
     const char \*p,  
     const char \*r);|  
 @apii{0,1,m}

Creates a copy of string `@id{s}`,  
 replacing any occurrence of the string `@id{p}`  
 with the string `@id{r}`.  
 Pushes the resulting string on the stack and returns it.

}

@APIEntry{lua\_Integer luaL\_len (lua\_State \*L, int index);|  
 @apii{0,0,e}

Returns the `@Q{length}` of the value at the given index  
 as a number;  
 it is equivalent to the  
`@Char{#}` operator in Lua `@see{len-op}`.  
 Raises an error if the result of the operation is not an integer.  
 (This case can only happen through metamethods.)

}

@APIEntry{  
 int luaL\_loadbuffer (lua\_State \*L,  
     const char \*buff,  
     size\_t sz,  
     const char \*name);|  
 @apii{0,1,-}

Equivalent to `@Lid{luaL_loadbufferx}` with `@id{mode}` equal to `@id{NULL}`.

}

```
@APIEntry{
int luaL_loadbufferx (lua_State *L,
                    const char *buff,
                    size_t sz,
                    const char *name,
                    const char *mode);|
@apii{0,1,-}
```

Loads a buffer as a Lua chunk.

This function uses @Lid{lua\_load} to load the chunk in the buffer pointed to by @id{buff} with size @id{sz}.

This function returns the same results as @Lid{lua\_load}.

@id{name} is the chunk name,

used for debug information and error messages.

The string @id{mode} works as in the function @Lid{lua\_load}.

```
}
```

```
@APIEntry{int luaL_loadfile (lua_State
*L, const char *filename);|
@apii{0,1,m}
```

Equivalent to @Lid{luaL\_loadfilex} with @id{mode} equal to @id{NULL}.

```
}
```

```
@APIEntry{int luaL_loadfilex (lua_State *L, const char *filename,
                             const char *mode);|
@apii{0,1,m}
```

Loads a file as a Lua chunk.

This function uses @Lid{lua\_load} to load the chunk in the file named @id{filename}.

If @id{filename} is @id{NULL},

then it loads from the standard input.

The first line in the file is ignored if it starts with a @T{#}.

The string @id{mode} works as in the function @Lid{lua\_load}.

This function returns the same results as @Lid{lua\_load}

or @Lid{LUA\_ERRFILE} for file-related errors.

As @Lid{lua\_load}, this function only loads the chunk;

it does not run it.

```
}
```



```
@APIEntry{int luaL_loadstring (lua_State *L, const char *s);|
@apii{0,1,-}}
```

Loads a string as a Lua chunk.

This function uses @Lid{lua\_load} to load the chunk in the zero-terminated string @id{s}.

This function returns the same results as @Lid{lua\_load}.

Also as @Lid{lua\_load}, this function only loads the chunk; it does not run it.

```
}
```

```
@APIEntry{void luaL_newlib (lua_State *L, const luaL_Reg l[]);|
@apii{0,1,m}}
```

Creates a new table and registers there the functions in the list @id{l}.

It is implemented as the following macro:

```
@verbatim{
(luaL_newlibtable(L,l), luaL_setfuncs(L,l,0))
}
```

The array @id{l} must be the actual array, not a pointer to it.

```
}
```

```
@APIEntry{void luaL_newlibtable (lua_State *L, const luaL_Reg l[]);|
@apii{0,1,m}}
```

Creates a new table with a size optimized to store all entries in the array @id{l} (but does not actually store them).

It is intended to be used in conjunction with @Lid{luaL\_setfuncs} @seeF{luaL\_newlib}.

It is implemented as a macro.

The array @id{l} must be the actual array, not a pointer to it.

```
}
```

```
@APIEntry{int luaL_newmetatable (lua_State *L, const char *tname);|
```

@apii{0,1,m}

If the registry already has the key @id{tname},  
returns 0.  
Otherwise,  
creates a new table to be used as a  
metatable for userdata,  
adds to this new table the pair @T{\_\_name = tname},  
adds to the registry the pair @T{[tname] = new table},  
and returns 1.

In both cases,  
the function pushes onto the stack the final value associated  
with @id{tname} in the registry.

}

@APIEntry{lua\_State \*luaL\_newstate (void);|  
@apii{0,0,-}}

Creates a new Lua state.  
It calls @Lid{lua\_newstate} with an  
allocator based on the @N{ISO C} allocation functions  
and then sets a warning function and a panic function @see{C-error}  
that print messages to the standard error output.

Returns the new state,  
or @id{NULL} if there is a @x{memory allocation error}.

}

@APIEntry{void luaL\_openlibs (lua\_State \*L);|  
@apii{0,0,e}}

Opens all standard Lua libraries into the given state.

}

@APIEntry{  
T luaL\_opt (L, func, arg, dflt);|  
@apii{0,0,-}}

This macro is defined as follows:

@verbatim{  
(lua\_isnoneornil(L,(arg)) ? (dflt) : func(L,(arg)))  
}

In words, if the argument @id{arg} is nil or absent,  
the macro results in the default

@id{dflt}.

Otherwise, it results in the result of calling @id{func} with the state @id{L} and the argument index @id{arg} as arguments.

Note that it evaluates the expression @id{dflt} only if needed.

}

```
@APIEntry{
lua_Integer luaL_optinteger (lua_State *L,
                           int arg,
                           lua_Integer d);|
@apii{0,0,v}
```

If the function argument @id{arg} is an integer (or it is convertible to an integer), returns this integer.

If this argument is absent or is @nil, returns @id{d}.

Otherwise, raises an error.

}

```
@APIEntry{
const char *luaL_optlstring (lua_State *L,
                           int arg,
                           const char *d,
                           size_t *l);|
@apii{0,0,v}
```

If the function argument @id{arg} is a string, returns this string.

If this argument is absent or is @nil, returns @id{d}.

Otherwise, raises an error.

If @id{l} is not @id{NULL}, fills its referent with the result's length. If the result is @id{NULL} (only possible when returning @id{d} and @T{d == NULL}), its length is considered zero.

This function uses @Lid{lua\_tolstring} to get its result, so all conversions and caveats of that function apply here.

}

```
@APIEntry{lua_Number luaL_optnumber (lua_State *L, int arg, lua_Number d);|
@apii{0,0,v}
```

If the function argument @id{arg} is a number,  
returns this number as a @id{lua\_Number}.

If this argument is absent or is @nil,  
returns @id{d}.

Otherwise, raises an error.

```
}
```

```
@APIEntry{
const char *luaL_optstring (lua_State *L,
                           int arg,
                           const char *d);|
@apii{0,0,v}
```

If the function argument @id{arg} is a string,  
returns this string.

If this argument is absent or is @nil,  
returns @id{d}.

Otherwise, raises an error.

```
}
```

```
@APIEntry{char *luaL_prepbuffer (luaL_Buffer *B);|
@apii{?,?,m}
```

Equivalent to @Lid{luaL\_prepbuffsize}  
with the predefined size @defid{LUAL\_BUFFERSIZE}.

```
}
```

```
@APIEntry{char *luaL_prepbuffsize (luaL_Buffer *B, size_t sz);|
@apii{?,?,m}
```

Returns

an address to a space of size @id{sz}

where you can copy a string to be added to buffer @id{B}

@seeC{luaL\_Buffer}.

After copying the string into this space you must call

@Lid{luaL\_addsize} with the size of the string to actually add  
it to the buffer.

```
}
```

```
@APIEntry{void luaL_pushfail (lua_State *L);|
@apii{0,1,-}
```

Pushes the @fail value onto the stack @see{libraries}.

}

```
@APIEntry{void luaL_pushresult (luaL_Buffer *B);|
@apii{?,1,m}
```

Finishes the use of buffer @id{B} leaving the final string on the top of the stack.

}

```
@APIEntry{void luaL_pushresultsz (luaL_Buffer *B, size_t sz);|
@apii{?,1,m}
```

Equivalent to the sequence @Lid{luaL\_addsize}, @Lid{luaL\_pushresult}.

}

```
@APIEntry{int luaL_ref (lua_State *L, int t);|
@apii{1,0,m}
```

Creates and returns a @def{reference},  
in the table at index @id{t},  
for the object on the top of the stack (and pops the object).

A reference is a unique integer key.

As long as you do not manually add integer keys into the table @id{t},

@Lid{luaL\_ref}

ensures the uniqueness of the key it returns.

You can retrieve an object referred by the reference @id{r}

by calling @T{lua\_rawgeti(L, t, r)}.

The function @Lid{luaL\_unref} frees a reference.

If the object on the top of the stack is @nil,

@Lid{luaL\_ref} returns the constant @defid{LUA\_REFNIL}.

The constant @defid{LUA\_NOREF} is guaranteed to be different  
from any reference returned by @Lid{luaL\_ref}.

}

```
@APIEntry{
typedef struct luaL_Reg {
  const char *name;
  lua_CFunction func;
} luaL_Reg;
|
```

Type for arrays of functions to be registered by

@Lid{luaL\_setfuncs}.

@id{name} is the function name and @id{func} is a pointer to the function.

Any array of @Lid{luaL\_Reg} must end with a sentinel entry in which both @id{name} and @id{func} are @id{NULL}.

}

@APIEntry{

void luaL\_requiref (lua\_State \*L, const char \*modname,  
lua\_CFunction openf, int glb);|

@apii{0,1,e}

If @T{package.loaded[modname]} is not true,  
calls the function @id{openf} with the string @id{modname}  
as an argument  
and sets the call result to @T{package.loaded[modname]},  
as if that function has been called through @Lid{require}.

If @id{glb} is true,  
also stores the module into the global @id{modname}.

Leaves a copy of the module on the stack.

}

@APIEntry{void luaL\_setfuncs (lua\_State \*L, const luaL\_Reg \*l, int nup);|  
@apii{nup,0,m}

Registers all functions in the array @id{l}  
@seeC{luaL\_Reg} into the table on the top of the stack  
(below optional upvalues, see next).

When @id{nup} is not zero,  
all functions are created with @id{nup} upvalues,  
initialized with copies of the @id{nup} values  
previously pushed on the stack  
on top of the library table.  
These values are popped from the stack after the registration.

A function with a @id{NULL} value represents a placeholder,  
which is filled with @false.

}

@APIEntry{void luaL\_setmetatable (lua\_State \*L, const char \*tname);|

@apii{0,0,-}

Sets the metatable of the object on the top of the stack  
as the metatable associated  
with name @id{tname}  
in the registry @seeC{luaL\_newmetatable}.

}

```
@APIEntry{
typedef struct luaL_Stream {
    FILE *f;
    lua_CFunction closef;
} luaL_Stream;
|
```

The standard representation for @x{file handles}  
used by the standard I/O library.

A file handle is implemented as a full userdata,  
with a metatable called @id{LUA\_FILEHANDLE}  
(where @id{LUA\_FILEHANDLE} is a macro with the actual metatable's name).  
The metatable is created by the I/O library  
@seeF{luaL\_newmetatable}.

This userdata must start with the structure @id{luaL\_Stream};  
it can contain other data after this initial structure.  
The field @id{f} points to the corresponding C stream  
(or it can be @id{NULL} to indicate an incompletely created handle).  
The field @id{closef} points to a Lua function  
that will be called to close the stream  
when the handle is closed or collected;  
this function receives the file handle as its sole argument and  
must return either a true value, in case of success,  
or a false value plus an error message,  
in case of error.  
Once Lua calls this field,  
it changes the field value to @id{NULL}  
to signal that the handle is closed.

}

```
@APIEntry{void *luaL_testudata (lua_State *L, int arg, const char *tname);|
@apii{0,0,m}
```

This function works like @Lid{luaL\_checkudata},  
except that, when the test fails,  
it returns @id{NULL} instead of raising an error.

```
}
```

```
@APIEntry{const char *luaL_tolstring (lua_State *L, int idx, size_t *len);|
@apii{0,1,e}
```

Converts any Lua value at the given index to a @N{C string}  
in a reasonable format.

The resulting string is pushed onto the stack and also  
returned by the function @see{constchar}.

If @id{len} is not @id{NULL},  
the function also sets @T{\*len} with the string length.

If the value has a metatable with a @idx{\_\_tostring} field,  
then @id{luaL\_tolstring} calls the corresponding metamethod  
with the value as argument,  
and uses the result of the call as its result.

```
}
```

```
@APIEntry{
void luaL_traceback (lua_State *L, lua_State *L1, const char *msg,

    int level);|
@apii{0,1,m}
```

Creates and pushes a traceback of the stack @id{L1}.

If @id{msg} is not @id{NULL}, it is appended  
at the beginning of the traceback.

The @id{level} parameter tells at which level  
to start the traceback.

```
}
```

```
@APIEntry{int luaL_typeerror (lua_State *L, int arg, const char *tname);|
@apii{0,0,v}
```

Raises a type error for the argument @id{arg}  
of the @N{C function} that called it,  
using a standard message;  
@id{tname} is a @Q{name} for the expected type.  
This function never returns.

```
}
```

```
@APIEntry{const char *luaL_typename (lua_State *L, int index);|
@apii{0,0,-}
```



Returns the name of the type of the value at the given index.

}

@APIEntry{void luaL\_unref (lua\_State \*L, int t, int ref);|  
@apii{0,0,-}}

Releases the reference @id{ref} from the table at index @id{t}  
@seeC{luaL\_unref}.

The entry is removed from the table,  
so that the referred object can be collected.

The reference @id{ref} is also freed to be used again.

If @id{ref} is @Lid{LUA\_NOREF} or @Lid{LUA\_REFNIL},  
@Lid{luaL\_unref}  
does nothing.

}

@APIEntry{void luaL\_where (lua\_State \*L, int lvl);|  
@apii{0,1,m}}

Pushes onto the stack a string identifying the current position  
of the control at level @id{lvl} in the call stack.

Typically this string has the following format:

@verbatim{  
@rep{chunkname}:@rep{currentline}:  
}

@N{Level 0} is the running function,  
@N{level 1} is the function that called the running function,  
etc.

This function is used to build a prefix for error messages.

}

}

}

@C{-----}  
@sect1{libraries| @title{The Standard Libraries}}

@simplesect{

The standard Lua libraries provide useful functions  
that are implemented @N{in C} through the @N{C API}.

Some of these functions provide essential services to the language (e.g., `@Lid{type}` and `@Lid{getmetatable}`); others provide access to outside services (e.g., I/O); and others could be implemented in Lua itself, but that for different reasons deserve an implementation in C (e.g., `@Lid{table.sort}`).

All libraries are implemented through the official `@N{C API}` and are provided as separate `@N{C modules}`. Unless otherwise noted, these library functions do not adjust its number of arguments to its expected parameters. For instance, a function documented as `@T{foo(arg)}` should not be called without an argument.

The notation `@fail` means a false value representing some kind of failure. (Currently, `@fail` is equal to `@nil`, but that may change in future versions. The recommendation is to always test the success of these functions with `@T{(not status)}`, instead of `@T{(status == nil)}`.)

Currently, Lua has the following standard libraries:

`@itemize{`

`@item{@link{predefined|basic library};}`

`@item{@link{corolib|coroutine library};}`

`@item{@link{packlib|package library};}`

`@item{@link{strlib|string manipulation};}`

`@item{@link{utf8|basic UTF-8 support};}`

`@item{@link{tablib|table manipulation};}`

`@item{@link{mathlib|mathematical functions}`  
`(sin, log, etc.);}`

`@item{@link{iolib|input and output};}`

`@item{@link{oslib|operating system facilities};}`

`@item{@link{debuglib|debug facilities}.}`

}

Except for the basic and the package libraries,  
each library provides all its functions as fields of a global table  
or as methods of its objects.

To have access to these libraries,  
the @N{C host} program should call the @Lid{luaL\_openlibs} function,  
which opens all standard libraries.

Alternatively,  
the host program can open them individually by using  
@Lid{luaL\_requiref} to call  
@defid{luaopen\_base} (for the basic library),  
@defid{luaopen\_package} (for the package library),  
@defid{luaopen\_coroutine} (for the coroutine library),  
@defid{luaopen\_string} (for the string library),  
@defid{luaopen\_utf8} (for the UTF-8 library),  
@defid{luaopen\_table} (for the table library),  
@defid{luaopen\_math} (for the mathematical library),  
@defid{luaopen\_io} (for the I/O library),  
@defid{luaopen\_os} (for the operating system library),  
and @defid{luaopen\_debug}  
(for the debug library).

These functions are declared in @defid{lualib.h}.

}

@sect2{predefined| @title{Basic Functions}}

The basic library provides core functions to Lua.  
If you do not include this library in your application,  
you should check carefully whether you need to provide  
implementations for some of its facilities.

@LibEntry{assert (v [, message])|

Raises an error if  
the value of its argument @id{v} is false (i.e., @nil or @false);  
otherwise, returns all its arguments.

In case of error,  
@id{message} is the error object;  
when absent, it defaults to @St{assertion failed!}

}

@LibEntry{collectgarbage ([opt [, arg]])|

This function is a generic interface to the garbage collector.  
It performs different functions according to its first argument, @id{opt}:  
@description{

@item{@St{collect}}|

Performs a full garbage-collection cycle.

This is the default option.

}

@item{@St{stop}}|

Stops automatic execution of the garbage collector.

The collector will run only when explicitly  
invoked,

until a call to restart it.

}

@item{@St{restart}}|

Restarts automatic execution of the garbage collector.

}

@item{@St{count}}|

Returns the total memory in use by Lua in Kbytes.

The value has a fractional part,

so that it multiplied by 1024

gives the exact number of bytes in use by Lua.

}

@item{@St{step}}|

Performs a garbage-collection step.

The step @Q{size} is controlled by @id{arg}.

With a zero value,

the collector will perform one basic (indivisible) step.

For non-zero values,

the collector will perform as if that amount of memory  
(in Kbytes) had been allocated by Lua.

Returns @true if the step finished a collection cycle.

}

@item{@St{isrunning}}|

Returns a boolean that tells whether the collector is running  
(i.e., not stopped).

}

@item{@St{incremental}}|

Change the collector mode to incremental.

This option can be followed by three numbers:

the garbage-collector pause,

the step multiplier,

and the step size @see{incmode}.

A zero means to not change that value.

}

@item{@St{generational}|  
Change  
the collector mode to generational.  
This option can be followed by two numbers:  
the garbage-collector minor multiplier  
and the major multiplier @see{genmode}.

A zero means to not change that value.

}

}

See @See{GC} for more details about garbage collection  
and some of these options.

This function should not be called by a finalizer.

}

@LibEntry{dofile ([filename])|  
Opens the named file and executes its content as a Lua chunk.  
When called without arguments,  
@id{dofile} executes the content of the standard input (@id{stdin}).  
Returns all values returned by the chunk.  
In case of errors, @id{dofile} propagates the error  
to its caller.  
(That is, @id{dofile} does not run in protected mode.)

}

@LibEntry{error (message [, level])|  
Raises an error @see{error} with @id{message} as the error object.  
This function never returns.

Usually, @id{error} adds some information about the error position  
at the beginning of the message, if the message is a string.

The @id{level}

argument specifies how to get the error position.

With @N{level 1} (the default), the error position is where the  
@id{error} function was called.

@N{Level 2} points the error to where the function  
that called @id{error} was called; and so on.

Passing a @N{level 0} avoids the addition of error position information  
to the message.

}

@LibEntry{\_G|

A global variable (not a function) that holds the @x{global environment} @see{globalenv}. Lua itself does not use this variable; changing its value does not affect any environment, nor vice versa.

}

@LibEntry{getmetatable (object)|

If @id{object} does not have a metatable, returns @nil. Otherwise, if the object's metatable has a @idx{\_\_metatable} field, returns the associated value. Otherwise, returns the metatable of the given object.

}

@LibEntry{ipairs (t)|

Returns three values (an iterator function, the table @id{t}, and 0) so that the construction

```
@verbatim{
for i,v in ipairs(t) do @rep{body} end
}
```

will iterate over the key@En{ }value pairs (@T{1,t[1]}), (@T{2,t[2]}), @ldots, up to the first absent index.

}

@LibEntry{load (chunk [, chunkname [, mode [, env]])|

Loads a chunk.

If @id{chunk} is a string, the chunk is this string. If @id{chunk} is a function, @id{load} calls it repeatedly to get the chunk pieces. Each call to @id{chunk} must return a string that concatenates with previous results. A return of an empty string, @nil, or no value signals the end of the chunk.

If there are no syntactic errors, @id{load} returns the compiled chunk as a function; otherwise, it returns @fail plus the error message.

When you load a main chunk,  
the resulting function will always have exactly one upvalue,  
the `@id{_ENV}` variable `@see{globalenv}`.  
However,  
when you load a binary chunk created from a function `@seeF{string.dump}`,  
the resulting function can have an arbitrary number of upvalues,  
and there is no guarantee that its first upvalue will be  
the `@id{_ENV}` variable.  
(A non-main function may not even have an `@id{_ENV}`  
upvalue.)

Regardless, if the resulting function has any upvalues,  
its first upvalue is set to the value of `@id{env}`,  
if that parameter is given,  
or to the value of the `@x{global environment}`.  
Other upvalues are initialized with `@nil`.  
All upvalues are fresh, that is,  
they are not shared with any other function.

`@id{chunkname}` is used as the name of the chunk for error messages  
and debug information `@see{debugI}`.  
When absent,  
it defaults to `@id{chunk}`, if `@id{chunk}` is a string,  
or to `@St{=(load)}` otherwise.

The string `@id{mode}` controls whether the chunk can be text or binary  
(that is, a precompiled chunk).  
It may be the string `@St{b}` (only `@x{binary chunk}`s),  
`@St{t}` (only text chunks),  
or `@St{bt}` (both binary and text).  
The default is `@St{bt}`.

It is safe to load malformed binary chunks;  
`@id{load}` signals an appropriate error.  
However,  
Lua does not check the consistency of the code inside binary chunks;  
running maliciously crafted bytecode can crash the interpreter.

}

`@LibEntry{loadfile`  
`[[filename [, mode [, env]]]]`

Similar to `@Lid{load}`,  
but gets the chunk from file `@id{filename}`  
or from the standard input,  
if no file name is given.

}

@LibEntry{next (table [, index])}

Allows a program to traverse all fields of a table.

Its first argument is a table and its second argument is an index in this table.

A call to @id{next} returns the next index of the table and its associated value.

When called with @nil as its second argument,

@id{next} returns an initial index and its associated value.

When called with the last index,

or with @nil in an empty table,

@id{next} returns @nil.

If the second argument is absent, then it is interpreted as @nil.

In particular,

you can use @T{next(t)} to check whether a table is empty.

The order in which the indices are enumerated is not specified,

@emph{even for numeric indices}.

(To traverse a table in numerical order,

use a numerical @Rw{for}.)

You should not assign any value to a non-existent field in a table during its traversal.

You

may however modify existing fields.

In particular, you may set existing fields to nil.

}

@LibEntry{pairs (t)}

If @id{t} has a metamethod @idx{\_\_pairs},

calls it with @id{t} as argument and returns the first three results from the call.

Otherwise,

returns three values: the @Lid{next} function, the table @id{t}, and @nil, so that the construction

@verbatim{

for k,v in pairs(t) do @rep{body} end

}

will iterate over all key@En{}value pairs of table @id{t}.

See function @Lid{next} for the caveats of modifying



the table during its traversal.

}

@LibEntry{pcall (f [, arg1, @Cdots])|

Calls the function @id{f} with  
the given arguments in @emphx{protected mode}.  
This means that any error @N{inside @T{f}} is not propagated;  
instead, @id{pcall} catches the error  
and returns a status code.

Its first result is the status code (a boolean),  
which is @true if the call succeeds without errors.  
In such case, @id{pcall} also returns all results from the call,  
after this first result.

In  
case of any error, @id{pcall} returns @false plus the error object.  
Note that errors caught by @id{pcall} do not call a message handler.

}

@LibEntry{print (@Cdots)|

Receives any number of arguments  
and prints their values to @id{stdout},  
converting each argument to a string  
following the same rules of @Lid{tostring}.

The function @id{print} is not intended for formatted output,  
but only as a quick way to show a value,  
for instance for debugging.  
For complete control over the output,  
use @Lid{string.format} and @Lid{io.write}.

}

@LibEntry{rawequal (v1, v2)|

Checks whether @id{v1} is equal to @id{v2},  
without invoking the @idx{\_\_eq} metamethod.  
Returns a boolean.

}

@LibEntry{rawget (table, index)|

Gets the real value of @T{table[index]},  
without using the @idx{\_\_index} metavalue.  
@id{table} must be a table;  
@id{index} may be any value.

}

@LibEntry{rawlen (v)|

Returns the length of the object @id{v},  
which must be a table or a string,  
without invoking the @idx{\_\_len} metamethod.  
Returns  
an integer.

}

@LibEntry{rawset (table, index, value)|

Sets the real value of @T{table[index]} to @id{value},  
without using the @idx{\_\_newindex} metavalue.  
@id{table} must be a table,  
@id{index} any value different from @nil and @x{NaN},  
and @id{value} any Lua value.

This function returns @id{table}.

}

@LibEntry{select (index, @Cdots)|

If @id{index} is a number,  
returns all arguments after argument number @id{index};  
a negative number indexes from the end (@num{-1} is the last argument).  
Otherwise, @id{index} must be the string @T{"#"},  
and @id{select} returns the total number of extra arguments it received.

}

@LibEntry{setmetatable (table, metatable)|

Sets the metatable for the given table.  
If @id{metatable} is @nil,  
removes the metatable of the given table.  
If the original metatable has a @idx{\_\_metatable} field,  
raises an error.

This function returns @id{table}.

To change the metatable of other types from Lua code,  
you must use the @link{debuglib|debug library}.

}

@LibEntry{tonumber

(e [, base])|

When called with no @id{base},  
@id{tonumber} tries to convert its argument to a number.  
If the argument is already a number or  
a string convertible to a number,  
then @id{tonumber} returns this number;  
otherwise, it returns @fail.

The conversion of strings can result in integers or floats,  
according to the lexical conventions of Lua @see{lexical}.  
The string may have leading and trailing spaces and a sign.

When called with @id{base},  
then @id{e} must be a string to be interpreted as  
an integer numeral in that base.  
The base may be any integer between 2 and 36, inclusive.  
In bases @N{above 10}, the letter @Char{A} (in either upper or lower case)  
@N{represents 10}, @Char{B} @N{represents 11}, and so forth,  
with @Char{Z} representing 35.  
If the string @id{e} is not a valid numeral in the given base,  
the function returns @fail.

}

@LibEntry{tostring (v)|

Receives a value of any type and  
converts it to a string in a human-readable format.

If the metatable  
of @id{v} has a @idx{\_\_tostring} field,  
then @id{tostring} calls the corresponding value  
with @id{v} as argument,  
and uses the result of the call as its result.  
Otherwise, if the metatable of @id{v} has a @idx{\_\_name} field  
with a string value,  
@id{tostring} may use that string in its final result.

For complete control of how numbers are converted,  
use @Lid{string.format}.

}

@LibEntry{type (v)|

Returns the type of its only argument, coded as a string.  
The possible results of this function are

@St{nil} (a string, not the value @nil),  
@St{number},  
@St{string},  
@St{boolean},  
@St{table},  
@St{function},  
@St{thread},  
and @St{userdata}.

}

@LibEntry{\_VERSION|

A global variable (not a function) that  
holds a string containing the running Lua version.  
The current value of this variable is @St{Lua 5.4}.

}

@LibEntry{warn (msg1, @Cdots)|

Emits a warning with a message composed by the concatenation  
of all its arguments (which should be strings).

By convention,  
a one-piece message starting  
with @Char{@At}  
is intended to be a @emph{control message},  
which is a message to the warning system itself.  
In particular, the standard warning function in Lua  
recognizes the control messages @St{@At{}off},  
to stop the emission of warnings,  
and @St{@At{}on}, to (re)start the emission;  
it ignores unknown control messages.

}

@LibEntry{xpcall (f, msg1 [, arg1, @Cdots])|

This function is similar to @Lid{pcall},  
except that it sets a new @x{message handler} @id{msg1}.

}

}

@sect2{corolib| @title{Coroutine Manipulation}}

This library comprises the operations to manipulate coroutines, which come inside the table `@defid{coroutine}`. See `@See{coroutine}` for a general description of coroutines.

`@LibEntry{coroutine.close (co)|`

Closes coroutine `@id{co}`, that is, closes all its pending to-be-closed variables and puts the coroutine in a dead state. The given coroutine must be dead or suspended. In case of error (either the original error that stopped the coroutine or errors in closing methods), returns `@false` plus the error object; otherwise returns `@true`.

`}`

`@LibEntry{coroutine.create (f)|`

Creates a new coroutine, with body `@id{f}`. `@id{f}` must be a function. Returns this new coroutine, an object with type `@T{"thread"}`.

`}`

`@LibEntry{coroutine.isyieldable ([co])|`

Returns `@true` when the coroutine `@id{co}` can yield. The default for `@id{co}` is the running coroutine.

A coroutine is yieldable if it is not the main thread and it is not inside a non-yieldable `@N{C function}`.

`}`

`@LibEntry{coroutine.resume (co [, val1, @Cdots])|`

Starts or continues the execution of coroutine `@id{co}`. The first time you resume a coroutine, it starts running its body. The values `@id{val1}`, `@ldots` are passed as the arguments to the body function. If the coroutine has yielded,

@id{resume} restarts it;  
the values @id{val1}, @ldots are passed  
as the results from the yield.

If the coroutine runs without any errors,  
@id{resume} returns @true plus any values passed to @id{yield}  
(when the coroutine  
yields) or any values returned by the body function  
(when the coroutine terminates).  
If there is any error,  
@id{resume} returns @false plus the error message.

}

@LibEntry{coroutine.running ()|

Returns the running coroutine plus a boolean,  
@true when the running coroutine is the main one.

}

@LibEntry{coroutine.status (co)|

Returns the status of the coroutine @id{co}, as a string:  
@T{"running"},  
if the coroutine is running  
(that is, it is the one that called @id{status});  
@T{"suspended"}, if the coroutine is suspended in a call to @id{yield},  
or if it has not started running yet;  
@T{"normal"} if the coroutine is active but not running  
(that is, it has resumed another coroutine);  
and @T{"dead"} if the coroutine has finished its body function,  
or if it has stopped with an error.

}

@LibEntry{coroutine.wrap (f)|

Creates a new coroutine, with body @id{f};  
@id{f} must be a function.  
Returns a function that resumes the coroutine each time it is called.  
Any arguments passed to this  
function behave as the  
extra arguments to @id{resume}.  
The function returns the same values returned by @id{resume},  
except the first boolean.  
In case of error,  
the function closes the coroutine and propagates the error.

```
}
```

```
@LibEntry{coroutine.yield (@Cdots)|
```

Suspends the execution of the calling coroutine.

Any arguments to @id{yield} are passed as extra results to @id{resume}.

```
}
```

```
}
```

```
@sect2{packlib| @title{Modules}}
```

The package library provides basic facilities for loading modules in Lua.

It exports one function directly in the global environment:

@Lid{require}.

Everything else is exported in the table @defid{package}.

```
@LibEntry{require (modname)|
```

Loads the given module.

The function starts by looking into the @Lid{package.loaded} table to determine whether @id{modname} is already loaded.

If it is, then @id{require} returns the value stored at @T{package.loaded[modname]}.

(The absence of a second result in this case signals that this call did not have to load the module.)

Otherwise,

it tries to find a @emph{loader} for the module.

To find a loader,

@id{require} is guided by the table @Lid{package.searchers}.

Each item in this table is a search function, that searches for the module in a particular way.

By changing this table,

we can change how @id{require} looks for a module.

The following explanation is based on the default configuration for @Lid{package.searchers}.

First @id{require} queries @T{package.preload[modname]}.

If it has a value,

this value (which must be a function) is the loader.

Otherwise @id{require} searches for a Lua loader using the path stored in @Lid{package.path}.

If that also fails, it searches for a @N{C loader} using the

path stored in `@Lid{package.cpath}`.

If that also fails,

it tries an `@emph{all-in-one}` loader `@seeF{package.searchers}`.

Once a loader is found,

`@id{require}` calls the loader with two arguments:

`@id{modname}` and an extra value,

a `@emph{loader data}`,

also returned by the searcher.

The loader data can be any value useful

to the module;

for the default searchers,

it indicates where the loader was found.

(For instance, if the loader came from a file,

this extra value is the file path.)

If the loader returns any non-nil value,

`@id{require}` assigns the returned value to `@T{package.loaded[modname]}`.

If the loader does not return a non-nil value and

has not assigned any value to `@T{package.loaded[modname]}`,

then `@id{require}` assigns `@true` to this entry.

In any case, `@id{require}` returns the

final value of `@T{package.loaded[modname]}`.

Besides that value, `@id{require}` also returns as a second result

the loader data returned by the searcher,

which indicates how `@id{require}` found the module.

If there is any error loading or running the module,

or if it cannot find any loader for the module,

then `@id{require}` raises an error.

}

`@LibEntry{package.config|`

A string describing some compile-time configurations for packages.

This string is a sequence of lines:

`@itemize{`

`@item{The first line is the @x{directory separator} string.`

Default is `@Char{\\}` for `@x{Windows}` and `@Char{/}` for all other systems.}

`@item{The second line is the character that separates templates in a path.`

Default is `@Char{;}.}`

`@item{The third line is the string that marks the substitution points in a template.`

Default is `@Char{?}.}`



@item{The fourth line is a string that, in a path in @x{Windows}, is replaced by the executable's directory.  
Default is @Char{!}.}

@item{The fifth line is a mark to ignore all text after it when building the @id{luaopen\_} function name.  
Default is @Char{-}.}

}

}

@LibEntry{package.cpath|

A string with the path used by @Lid{require} to search for a @N{C loader}.

Lua initializes the @N{C path} @Lid{package.cpath} in the same way it initializes the Lua path @Lid{package.path}, using the environment variable @defid{LUA\_CPATH\_5\_4}, or the environment variable @defid{LUA\_CPATH}, or a default path defined in @id{luaconf.h}.

}

@LibEntry{package.loaded|

A table used by @Lid{require} to control which modules are already loaded.  
When you require a module @id{modname} and @T{package.loaded[modname]} is not false, @Lid{require} simply returns the value stored there.

This variable is only a reference to the real table; assignments to this variable do not change the table used by @Lid{require}.  
The real table is stored in the C registry @see{registry}, indexed by the key @defid{LUA\_LOADED\_TABLE}, a string.

}

@LibEntry{package.loadlib (libname, funcname)|

Dynamically links the host program with the @N{C library} @id{libname}.

If @id{funcname} is @St{\*},

then it only links with the library,  
making the symbols exported by the library  
available to other dynamically linked libraries.  
Otherwise,  
it looks for a function @id{funcname} inside the library  
and returns this function as a @N{C function}.  
So, @id{funcname} must follow the @Lid{lua\_CFunction} prototype  
@seeC{lua\_CFunction}.

This is a low-level function.  
It completely bypasses the package and module system.  
Unlike @Lid{require},  
it  
does not perform any path searching and  
does not automatically adds extensions.  
@id{libname} must be the complete file name of the @N{C library},  
including if necessary a path and an extension.  
@id{funcname} must be the exact name exported by the @N{C library}  
(which may depend on the @N{C compiler} and linker used).

This functionality is not supported by @N{ISO C}.  
As such, it is only available on some platforms  
(Windows, Linux, Mac OS X, Solaris, BSD,  
plus other Unix systems that support the @id{dlfcn} standard).

This function is inherently insecure,  
as it allows Lua to call any function in any readable dynamic  
library in the system.  
(Lua calls any function assuming the function  
has a proper prototype and respects a proper protocol  
@see{lua\_CFunction}.  
Therefore,  
calling an arbitrary function in an arbitrary dynamic library  
more often than not results in an access violation.)

}

@LibEntry{package.path|

A string with the path used by @Lid{require}  
to search for a Lua  
loader.

At start-up, Lua initializes this variable with  
the value of the environment variable @defid{LUA\_PATH\_5\_4} or  
the environment variable @defid{LUA\_PATH} or  
with a default path defined in @id{luaconf.h},  
if those environment variables are not defined.

A `@St{;;}` in the value of the environment variable  
is replaced by the default path.

}

`@LibEntry{package.preload|`

A table to store loaders for specific modules  
`@seeF{require}`.

This variable is only a reference to the real table;  
assignments to this variable do not change the  
table used by `@Lid{require}`.  
The real table is stored in the C registry `@see{registry}`,  
indexed by the key `@defid{LUA_PRELOAD_TABLE}`, a string.

}

`@LibEntry{package.searchers|`

A table used by `@Lid{require}` to control how to find modules.

Each entry in this table is a `@def{searcher function}`.  
When looking for a module,  
`@Lid{require}` calls each of these searchers in ascending order,  
with the module name (the argument given to `@Lid{require}`) as its  
sole argument.

If  
the searcher finds the module,  
it returns another function, the module `@def{loader}`,  
plus an extra value, a `@emph{loader data}`,  
that will be passed to that loader and  
returned as a second result by `@Lid{require}`.  
If it cannot find the module,  
it returns a string explaining why  
(or `@nil` if it has nothing to say).

Lua initializes this table with four searcher functions.

The first searcher simply looks for a loader in the  
`@Lid{package.preload}` table.

The second searcher looks for a loader as a Lua library,  
using the path stored at `@Lid{package.path}`.  
The search is done as described in function `@Lid{package.searchpath}`.

The third searcher looks for a loader as a `@N{C library}`,  
using the path given by the variable `@Lid{package.cpath}`.

Again,  
the search is done as described in function `@Lid{package.searchpath}`.  
For instance,  
if the `@N{C path}` is the string  
`@verbatim{`  
`"/?.so;/??.dll;/usr/local/?/init.so"`  
`}`  
the searcher for module `@id{foo}`  
will try to open the files `@T{./foo.so}`,  
`@T{./foo.dll}`,  
and `@T{/usr/local/foo/init.so}`, in that order.  
Once it finds a `@N{C library}`,  
this searcher first uses a dynamic link facility to link the  
application with the library.  
Then it tries to find a `@N{C function}` inside the library to  
be used as the loader.  
The name of this `@N{C function}` is the string `@St{luaopen_}`  
concatenated with a copy of the module name where each dot  
is replaced by an underscore.  
Moreover, if the module name has a hyphen,  
its suffix after (and including) the first hyphen is removed.  
For instance, if the module name is `@id{a.b.c-v2.1}`,  
the function name will be `@id{luaopen_a_b_c}`.

The fourth searcher tries an `@def{all-in-one loader}`.  
It searches the `@N{C path}` for a library for  
the root name of the given module.  
For instance, when requiring `@id{a.b.c}`,  
it will search for a `@N{C library}` for `@id{a}`.  
If found, it looks into it for an open function for  
the submodule;  
in our example, that would be `@id{luaopen_a_b_c}`.  
With this facility, a package can pack  
several `@N{C submodules}`  
into one single library,  
with each submodule keeping its original open function.

All searchers except the first one (preload) return as the extra value  
the file path where the module was found,  
as returned by `@Lid{package.searchpath}`.  
The first searcher always returns the string `@St{:preload:}`.

Searchers should raise no errors and have no side effects in Lua.  
(They may have side effects in C,  
for instance by linking the application with a library.)

}

@LibEntry{package.searchpath (name, path [, sep [, rep]])|

Searches for the given @id{name} in the given @id{path}.

A path is a string containing a sequence of  
@emph{templates} separated by semicolons.  
For each template,  
the function replaces each interrogation mark (if any)  
in the template with a copy of @id{name}  
wherein all occurrences of @id{sep}  
(a dot, by default)  
were replaced by @id{rep}  
(the system's directory separator, by default),  
and then tries to open the resulting file name.

For instance, if  
the path is the string  
@verbatim{  
"./?.lua;./?.lc;/usr/local/?/init.lua"  
}  
the search for the name @id{foo.a}  
will try to open the files  
@T{./foo/a.lua}, @T{./foo/a.lc}, and  
@T{/usr/local/foo/a/init.lua}, in that order.

Returns the resulting name of the first file that it can  
open in read mode (after closing the file),  
or @fail plus an error message if none succeeds.  
(This error message lists all file names it tried to open.)

}

}

@sect2{strlib| @title{String Manipulation}}

@simplesect{

This library provides generic functions for string manipulation,  
such as finding and extracting substrings, and pattern matching.  
When indexing a string in Lua, the first character is at @N{position 1}  
(not @N{at 0}, as in C).  
Indices are allowed to be negative and are interpreted as indexing backwards,  
from the end of the string.  
Thus, the last character is at position @num{-1}, and so on.

The string library provides all its functions inside the table  
@defid{string}.

It also sets a `@x{metatable for strings}` where the `@idx{__index}` field points to the `@id{string}` table. Therefore, you can use the string functions in object-oriented style. For instance, `@T{string.byte(s,i)}` can be written as `@T{s:byte(i)}`.

The string library assumes one-byte character encodings.

`@LibEntry{string.byte (s [, i [, j]])}`  
Returns the internal numeric codes of the characters `@T{s[i]}`, `@T{s[i+1]}`, `@ldots`, `@T{s[j]}`.  
The default value for `@id{i}` `@N{is 1}`;  
the default value for `@id{j}` `@N{is @id{i}}`.  
These indices are corrected following the same rules of function `@Lid{string.sub}`.

Numeric codes are not necessarily portable across platforms.

}

`@LibEntry{string.char (@Cdots)}`  
Receives zero or more integers.  
Returns a string with length equal to the number of arguments, in which each character has the internal numeric code equal to its corresponding argument.

Numeric codes are not necessarily portable across platforms.

}

`@LibEntry{string.dump (function [, strip])}`

Returns a string containing a binary representation  
(a `@emph{binary chunk}`)  
of the given function,  
so that a later `@Lid{load}` on this string returns  
a copy of the function (but with new upvalues).  
If `@id{strip}` is a true value,  
the binary representation may not include all debug information  
about the function,  
to save space.

Functions with upvalues have only their number of upvalues saved.  
When (re)loaded,  
those upvalues receive fresh instances.

(See the `@Lib{load}` function for details about how these upvalues are initialized. You can use the debug library to serialize and reload the upvalues of a function in a way adequate to your needs.)

}

`@LibEntry{string.find (s, pattern [, init [, plain]])|`

Looks for the first match of

`@id{pattern}` `@see{pm}` in the string `@id{s}`.

If it finds a match, then `@id{find}` returns the indices `@N{of @T{s}}` where this occurrence starts and ends;

otherwise, it returns `@fail`.

A third, optional numeric argument `@id{init}` specifies where to start the search;

its default value `@N{is 1}` and can be negative.

A `@true` as a fourth, optional argument `@id{plain}` turns off the pattern matching facilities, so the function does a plain `@Q{find substring}` operation, with no characters in `@id{pattern}` being considered magic.

If the pattern has captures, then in a successful match the captured values are also returned, after the two indices.

}

`@LibEntry{string.format (formatstring, @Cdots)|`

Returns a formatted version of its variable number of arguments following the description given in its first argument, which must be a string.

The format string follows the same rules as the `@ANSI{sprintf}`.

The only differences are that the conversion specifiers and modifiers `@id{F}`, `@id{n}`, `@T{*}`, `@id{h}`, `@id{L}`, and `@id{l}` are not supported and that there is an extra specifier, `@id{q}`.

Both width and precision, when present, are limited to two digits.

The specifier `@id{q}` formats booleans, nil, numbers, and strings in a way that the result is a valid constant in Lua source code.

Booleans and nil

are written in the obvious way (`@id{true}`, `@id{false}`, `@id{nil}`).

Floats are written in hexadecimal,

to preserve full precision.

A string is written between double quotes,  
using escape sequences when necessary to ensure that  
it can safely be read back by the Lua interpreter.

For instance, the call

```
@verbatim{
string.format("%q", 'a string with "quotes" and \n new line')
}
```

may produce the string:

```
@verbatim{
"a string with \"quotes\" and \
new line"
}
```

This specifier does not support modifiers (flags, width, precision).

The conversion specifiers

@id{A}, @id{a}, @id{E}, @id{e}, @id{f},  
@id{G}, and @id{g} all expect a number as argument.

The specifiers @id{c}, @id{d},  
@id{i}, @id{o}, @id{u}, @id{X}, and @id{x}  
expect an integer.

When Lua is compiled with a C89 compiler,  
the specifiers @id{A} and @id{a} (hexadecimal floats)  
do not support modifiers.

The specifier @id{s} expects a string;

if its argument is not a string,

it is converted to one following the same rules of @Lid{tostring}.

If

the specifier has any modifier,

the corresponding string argument should not contain @x{embedded zeros}.

The specifier @id{p} formats the pointer  
returned by @Lid{lua\_topointer}.

That gives a unique string identifier for tables, userdata,  
threads, strings, and functions.

For other values (numbers, nil, booleans),  
this specifier results in a string representing  
the pointer @id{NULL}.

}

@LibEntry{string.gmatch (s, pattern [, init])|

Returns an iterator function that,

each time it is called,

returns the next captures from @id{pattern} @see{pm}

over the string @id{s}.



If `@id{pattern}` specifies no captures, then the whole match is produced in each call. A third, optional numeric argument `@id{init}` specifies where to start the search; its default value `@N{is 1}` and can be negative.

As an example, the following loop will iterate over all the words from string `@id{s}`, printing one per line:

```
@verbatim{
s = "hello world from Lua"
for w in string.gmatch(s, "%a+") do
  print(w)
end
}
```

The next example collects all pairs `@T{key=value}` from the given string into a table:

```
@verbatim{
t = {}
s = "from=world, to=Lua"
for k, v in string.gmatch(s, "(%w+)=(%w+)") do
  t[k] = v
end
}
```

For this function, a caret `@Char{^}` at the start of a pattern does not work as an anchor, as this would prevent the iteration.

```
}
```

```
@LibEntry{string.gsub (s, pattern, repl [, n])|
```

Returns a copy of `@id{s}`

in which all (or the first `@id{n}`, if given)

occurrences of the `@id{pattern}` `@see{pm}` have been replaced by a replacement string specified by `@id{repl}`,

which can be a string, a table, or a function.

`@id{gsub}` also returns, as its second value, the total number of matches that occurred.

The name `@id{gsub}` comes from `@emph{Global SUBstitution}`.

If `@id{repl}` is a string, then its value is used for replacement.

The `@N{character @T{%}}` works as an escape character:

any sequence in `@id{repl}` of the form `@T{%@rep{d}}`,

with `@rep{d}` between 1 and 9,

stands for the value of the `@rep{d}`-th captured substring;

the

sequence `@T{%0}` stands for the whole match;  
the sequence `@T{%%}` stands for a `@N{single @T{%}}`.

If `@id{repl}` is a table, then the table is queried for every match,  
using the first capture as the key.

If `@id{repl}` is a function, then this function is called every time a  
match occurs, with all captured substrings passed as arguments,  
in order.

In any case,  
if the pattern specifies no captures,  
then it behaves as if the whole pattern was inside a capture.

If the value returned by the table query or by the function call  
is a string or a number,  
then it is used as the replacement string;  
otherwise, if it is `@Rw{false}` or `@nil`,  
then there is no replacement  
(that is, the original match is kept in the string).

Here are some examples:

```
@verbatim{
```

```
x = string.gsub("hello world", "(%w+)", "%1 %1")
```

```
--> x="hello hello world world"
```

```
x = string.gsub("hello world", "%w+", "%0 %0", 1)
```

```
--> x="hello hello world"
```

```
x = string.gsub("hello world from Lua", "(%w+)%s*(%w+)", "%2 %1")
```

```
--> x="world hello Lua
```

```
from"
```

```
x = string.gsub("home = $HOME, user = $USER", "%$(%w+)", os.getenv)
```

```
--> x="home = /home/roberto, user = roberto"
```

```
x = string.gsub("4+5 = $return 4+5$", "%$(.)%", function (s)
```

```
    return load(s)()
```

```
end)
```

```
--> x="4+5 = 9"
```

```
local t = {name="lua", version="5.4"}
```

```
x = string.gsub("$name-$version.tar.gz", "%$(%w+)", t)
```

```
--> x="lua-5.4.tar.gz"
```

```
}
```

```
}
```

@LibEntry{string.len (s)|

Receives a string and returns its length.  
The empty string @T{""} has length 0.  
Embedded zeros are counted,  
so @T{"a\000bc\000"} has length 5.

}

@LibEntry{string.lower (s)|

Receives a string and returns a copy of this string with all  
uppercase letters changed to lowercase.  
All other characters are left unchanged.  
The definition of what an uppercase letter is depends on the current locale.

}

@LibEntry{string.match (s, pattern [, init])|

Looks for the first @emph{match} of  
the @id{pattern} @see{pm} in the string @id{s}.  
If it finds one, then @id{match} returns  
the captures from the pattern;  
otherwise it  
returns @fail.  
If @id{pattern} specifies no captures,  
then the whole match is returned.  
A third, optional numeric argument @id{init} specifies  
where to start the search;  
its default value @N{is 1} and can be negative.

}

@LibEntry{string.pack (fmt, v1, v2, @Cdots)|

Returns a binary string containing the values @id{v1}, @id{v2}, etc.  
serialized in binary form (packed)  
according to the format string @id{fmt} @see{pack}.

}

@LibEntry{string.packsize (fmt)|

Returns the length of a string resulting from @Lid{string.pack}  
with the given format.  
The format string cannot have the variable-length options  
@Char{s} or @Char{z} @see{pack}.

}

@LibEntry{string.rep (s, n [, sep])|

Returns a string that is the concatenation of @id{n} copies of the string @id{s} separated by the string @id{sep}.

The default value for @id{sep} is the empty string (that is, no separator).

Returns the empty string if @id{n} is not positive.

(Note that it is very easy to exhaust the memory of your machine with a single call to this function.)

}

@LibEntry{string.reverse (s)|

Returns a string that is the string @id{s} reversed.

}

@LibEntry{string.sub (s, i [, j])|

Returns the substring of @id{s} that starts at @id{i} and continues until @id{j};

@id{i} and @id{j} can be negative.

If @id{j} is absent, then it is assumed to be equal to @num{-1} (which is the same as the string length).

In particular,

the call @T{string.sub(s,1,j)} returns a prefix of @id{s} with length @id{j},

and @T{string.sub(s, -i)} (for a positive @id{i}) returns a suffix of @id{s} with length @id{i}.

If, after the translation of negative indices,

@id{i} is less than 1,

it is corrected to 1.

If @id{j} is greater than the string length, it is corrected to that length.

If, after these corrections,

@id{i} is greater than @id{j},

the function returns the empty string.

}

@LibEntry{string.unpack (fmt, s [, pos])|

Returns the values packed in string @id{s} @seeF{string.pack}  
according to the format string @id{fmt} @see{pack}.

An

optional @id{pos} marks where  
to start reading in @id{s} (default is 1).

After the read values,  
this function also returns the index of the first unread byte in @id{s}.

}

@LibEntry{string.upper (s)|

Receives a string and returns a copy of this string with all  
lowercase letters changed to uppercase.  
All other characters are left unchanged.  
The definition of what a lowercase letter is depends on the current locale.

}

}

@sect3{pm| @title{Patterns}}

@simplesect{

Patterns in Lua are described by regular strings,  
which are interpreted as patterns by the pattern-matching functions  
@Lid{string.find},  
@Lid{string.gmatch},  
@Lid{string.gsub},  
and @Lid{string.match}.  
This section describes the syntax and the meaning  
(that is, what they match) of these strings.

}

@sect4{@title{Character Class:}}

A @def{character class} is used to represent a set of characters.  
The following combinations are allowed in describing a character class:  
@description{

@item{@rep{x}|  
(where @rep{x} is not one  
of the @emphx{magic characters}  
@T{^\$()%%.[]\*+-.?})

represents the character `@emph{x}` itself.

}

`@item{@T{.}|` (a dot) represents all characters.}

`@item{@T{%a}|` represents all letters.}

`@item{@T{%c}|` represents all control characters.}

`@item{@T{%d}|` represents all digits.}

`@item{@T{%g}|` represents all printable characters except space.}

`@item{@T{%l}|` represents all lowercase letters.}

`@item{@T{%p}|` represents all punctuation characters.}

`@item{@T{%s}|` represents all space characters.}

`@item{@T{%u}|` represents all uppercase letters.}

`@item{@T{%w}|` represents all alphanumeric characters.}

`@item{@T{%x}|` represents all hexadecimal digits.}

`@item{@T{%@rep{x}}|` (where `@rep{x}` is any non-alphanumeric character)  
represents the character `@rep{x}`.

This is the standard way to escape the magic characters.

Any non-alphanumeric character

(including all punctuation characters, even the non-magical)

can be preceded by a `@Char{%}` to represent itself in a pattern.

}

`@item{@T{[@rep{set}]}`

represents

the class which is the union of all

characters in `@rep{set}`.

A range of characters can be specified by

separating the end characters of the range,

in ascending order, with a `@Char{-}`.

All classes `@T{%}@emph{x}` described above can also be used as  
components in `@rep{set}`.

All other characters in `@rep{set}` represent themselves.

For example, `@T{[%w_]}` (or `@T{[_%w]}`)

represents all alphanumeric characters plus the underscore,

`@T{[0-7]}` represents the octal digits,

and `@T{[0-7%l%-]}` represents the octal digits plus

the lowercase letters plus the `@Char{-}` character.

You can put a closing square bracket in a set  
by positioning it as the first character in the set.  
You can put a hyphen in a set  
by positioning it as the first or the last character in the set.  
(You can also use an escape for both cases.)

The interaction between ranges and classes is not defined.  
Therefore, patterns like `@T{[%a-z]}` or `@T{[a-%]}`  
have no meaning.  
}

`@item{@T{[^@rep{set}]}`  
represents the complement of `@rep{set}`,  
where  
`@rep{set}` is interpreted as above.  
}

}

For all classes represented by single letters (`@T{%a}`, `@T{%c}`, etc.),  
the corresponding uppercase letter represents the complement of the class.  
For instance, `@T{%S}` represents all non-space characters.

The definitions of letter, space, and other character groups  
depend on the current locale.  
In particular, the class `@T{[a-z]}` may not be equivalent to `@T{%l}`.  
}

`@sect4{@title{Pattern Item:}`  
A `@def{pattern item}` can be  
`@itemize{`

`@item{`  
a single character class,  
which matches any single character in the class;  
}

`@item{`  
a single character class followed by `@Char{*}`,  
which matches sequences of zero or more characters in the class.  
These repetition items will always match the longest possible sequence;  
}

`@item{`  
a single character class followed by `@Char{+}`,  
which matches sequences of one or more characters in the class.  
These repetition items will always match the longest possible sequence;

}

@item{

a single character class followed

by @Char{-},

which also matches sequences of zero or more characters in the class.

Unlike @Char{\*},

these repetition items will always match the shortest possible sequence;

}

@item{

a single character class followed by @Char{?},

which matches zero or one occurrence of a character in the class.

It always matches one occurrence if possible;

}

@item{

@T{%@rep{n}}, for @rep{n} between 1 and 9;

such item matches a substring equal to the @rep{n}-th captured string

(see below);

}

@item{

@T{%b@rep{xy}}, where @rep{x} and @rep{y} are two distinct characters;

such item matches strings that start @N{with @rep{x}}, end @N{with @rep{y}},

and where the @rep{x} and @rep{y} are @emph{balanced}.

This means that, if one reads the string from left to right,

counting @M{+1} for an @rep{x} and @M{-1} for a @rep{y},

the ending @rep{y} is the first @rep{y} where the count reaches 0.

For instance, the item @T{%b()} matches expressions with

balanced parentheses.

}

@item{

@T{%f[@rep{set}]}, a @def{frontier

pattern};

such item matches an empty string at any position such that

the next character belongs to @rep{set}

and the previous character does not belong to @rep{set}.

The set @rep{set} is interpreted as previously described.

The beginning and the end of the subject are handled as if

they were the character @Char{\0}.

}

}

}



```
@sect4{@title{Pattern:}}
```

A `@def{pattern}` is a sequence of pattern items.

A caret `@Char{^}` at the beginning of a pattern anchors the match at the beginning of the subject string.

A `@Char{$}` at the end of a pattern anchors the match at the end of the subject string.

At other positions,

`@Char{^}` and `@Char{$}` have no special meaning and represent themselves.

```
}
```

```
@sect4{@title{Captures:}}
```

A pattern can contain sub-patterns enclosed in parentheses; they describe `@def{captures}`.

When a match succeeds, the substrings of the subject string that match captures are stored (`@emph{captured}`) for future use.

Captures are numbered according to their left parentheses.

For instance,

in the pattern `@T{"(a*(.)%w(%s*)}"`,  
the part of the string matching `@T{"a*(.)%w(%s*)}"` is  
stored as the first capture, and therefore has `@N{number 1}`;  
the character matching `@St{.}` is captured with `@N{number 2}`,  
and the part matching `@St{%s*}` has `@N{number 3}`.

As a special case, the capture `@T{()}` captures the current string position (a number).

For instance, if we apply the pattern `@T{"()aa()}"` on the string `@T{"flaaap"}`, there will be two captures: `@N{3}` and `@N{5}`.

```
}
```

```
@sect4{@title{Multiple matches:}}
```

The function `@Lid{string.gsub}` and the iterator `@Lid{string.gmatch}` match multiple occurrences of the given pattern in the subject.

For these functions,

a new match is considered valid only if it ends at least one byte after the end of the previous match.

In other words, the pattern machine never accepts the empty string as a match immediately after another match.

As an example,

consider the results of the following code:

```
@verbatim{
```

```
> string.gsub("abc", "()a*()", print);
```

```
--> 1 2
```

```
--> 3
```

```
3
```

```
--> 4 4
```

```
}
```

The second and third results come from Lua matching an empty string after `@Char{b}` and another one after `@Char{c}`.  
 Lua does not match an empty string after `@Char{a}`, because it would end at the same position of the previous match.

}

}

`@sect3{pack| @title{Format Strings for Pack and Unpack}}`

The first argument to `@Lid{string.pack}`,  
`@Lid{string.packsize}`, and `@Lid{string.unpack}`  
 is a format string,  
 which describes the layout of the structure being created or read.

A format string is a sequence of conversion options.

The conversion options are as follows:

```
@description{
@item{@T{<}|sets little endian}
@item{@T{>}|sets big endian}
@item{@T{=}|sets native endian}
@item{@T{![@rep{n}]}|sets maximum alignment to @id{n}
(default is native alignment)}
@item{@T{b}|a signed byte (@id{char})}
@item{@T{B}|an unsigned byte (@id{char})}
@item{@T{h}|a signed @id{short} (native size)}
@item{@T{H}|an unsigned @id{short} (native size)}
@item{@T{l}|a signed @id{long} (native size)}
@item{@T{L}|an
  unsigned @id{long} (native size)}
@item{@T{j}|a @id{lua_Integer}}
@item{@T{J}|a @id{lua_Unsigned}}
@item{@T{T}|a @id{size_t} (native size)}
@item{@T{i[@rep{n}]}|a signed @id{int} with @id{n} bytes
(default is native size)}
@item{@T{I[@rep{n}]}|an unsigned @id{int} with @id{n} bytes
(default is native size)}
@item{@T{f}|a @id{float} (native size)}
@item{@T{d}|a @id{double} (native size)}
@item{@T{n}|a @id{lua_Number}}
@item{@T{c[@rep{n}]}|a fixed-sized string with @id{n} bytes}
@item{@T{z}|a zero-terminated string}
@item{@T{s[@emph{n}]}|a string preceded by its length
coded as an unsigned integer with @id{n} bytes
(default is a @id{size_t})}
@item{@T{x}|one byte of padding}
@item{@T{X[@rep{op}]}|an empty item that aligns
```

according to option `@id{op}`  
 (which is otherwise ignored)}  
`@item{@Char{ }(space) ignored}`  
`}`  
 (A `@St{[@rep{n}]}` means an optional integral numeral.)  
 Except for padding, spaces, and configurations  
 (options `@St{xX <=>!}`),  
 each option corresponds to an argument in `@Lid{string.pack}`  
 or  
 a result in `@Lid{string.unpack}`.

For options `@St{!@rep{n]}`, `@St{s@rep{n]}`, `@St{i@rep{n]}`, and `@St{I@rep{n]}`,  
`@id{n}` can be any integer between 1 and 16.  
 All integral options check overflows;  
`@Lid{string.pack}` checks whether the given value fits in the given size;  
`@Lid{string.unpack}` checks whether the read value fits in a Lua integer.  
 For the unsigned options,  
 Lua integers are treated as unsigned values too.

Any format string starts as if prefixed by `@St{!1=}`,  
 that is,  
 with maximum alignment of 1 (no alignment)  
 and native endianness.

Native endianness assumes that the whole system is  
 either big or little endian.  
 The packing functions will not emulate correctly the behavior  
 of mixed-endian formats.

Alignment works as follows:  
 For each option,  
 the format gets extra padding until the data starts  
 at an offset that is a multiple of the minimum between the  
 option size and the maximum alignment;  
 this minimum must be a power of 2.  
 Options `@St{c}` and `@St{z}` are not aligned;  
 option  
`@St{s}` follows the alignment of its starting integer.

All padding is filled with zeros by `@Lid{string.pack}`  
 and ignored by `@Lid{string.unpack}`.

`}`

`}`

`@sect2{utf8| @title{UTF-8 Support}`

This library provides basic support for `@x{UTF-8}` encoding.  
It provides all its functions inside the table `@defid{utf8}`.  
This library does not provide any support for `@x{Unicode}` other than the handling of the encoding.  
Any operation that needs the meaning of a character, such as character classification, is outside its scope.

Unless stated otherwise,  
all functions that expect a byte position as a parameter assume that the given position is either the start of a byte sequence or one plus the length of the subject string.  
As in the string library,  
negative indices count from the end of the string.

Functions that create byte sequences  
accept all values up to `@T{0x7FFFFFFF}`,  
as defined in the original UTF-8 specification;  
that implies byte sequences of up to six bytes.

Functions that interpret  
byte sequences only accept  
valid sequences (well formed and not overlong).  
By default, they only accept byte sequences  
that result in valid Unicode code points,  
rejecting values greater than `@T{10FFFF}` and surrogates.  
A boolean argument `@id{lax}`, when available,  
lifts these checks,  
so that all values up to `@T{0x7FFFFFFF}` are accepted.  
(Not well formed and overlong sequences are still rejected.)

`@LibEntry{utf8.char (@Cdots)|`

Receives zero or more integers,  
converts each one to its corresponding UTF-8 byte sequence  
and returns a string with the concatenation of all these sequences.

`}`

`@LibEntry{utf8.charpattern|`

The pattern (a string, not a function) `@St{[\0-\x7F\xC2-\xFD][\x80-\xBF]*}`  
`@see{pm}`,  
which matches exactly one UTF-8 byte sequence,  
assuming that the subject is a valid UTF-8 string.

`}`

@LibEntry{utf8.codes (s [, lax])|

Returns values so that the construction

@verbatim{

for p, c in utf8.codes(s) do @rep{body} end

}

will iterate over all UTF-8 characters in string @id{s},

with

@id{p} being the position (in bytes) and @id{c} the code point of each character.

It raises an error if it meets any invalid byte sequence.

}

@LibEntry{utf8.codepoint (s [, i [, j [, lax]])]|

Returns the code points (as integers) from all characters in @id{s}

that start between byte position @id{i} and @id{j} (both included).

The default for @id{i} is 1 and for @id{j} is @id{i}.

It raises an error if it meets any invalid byte sequence.

}

@LibEntry{utf8.len (s [, i [, j [, lax]])]|

Returns the number of UTF-8 characters in string @id{s}

that start between positions @id{i} and @id{j} (both inclusive).

The default for @id{i} is @num{1} and for @id{j} is @num{-1}.

If it finds any invalid byte sequence,

returns @fail plus the position of the first invalid byte.

}

@LibEntry{utf8.offset (s, n [, i)]|

Returns the position (in bytes) where the encoding of the

@id{n}-th character of @id{s}

(counting from position @id{i}) starts.

A negative @id{n} gets characters before position @id{i}.

The

default for @id{i} is 1 when @id{n} is non-negative

and @T{#s + 1} otherwise,

so that @T{utf8.offset(s, -n)} gets the offset of the

@id{n}-th character from the end of the string.

If the specified character is neither in the subject

nor right after its end,

the function returns @fail.

As a special case,  
when @id{n} is 0 the function returns the start of the encoding  
of the character that contains the @id{i}-th byte of @id{s}.

This function assumes that @id{s} is a valid UTF-8 string.

}

}

@sect2{tablib| @title{Table Manipulation}

This library provides generic functions for table manipulation.  
It provides all its functions inside the table @defid{table}.

Remember that, whenever an operation needs the length of a table,  
all caveats about the length operator apply @see{len-op}.  
All functions ignore non-numeric keys  
in the tables given as arguments.

@LibEntry{table.concat (list [, sep [, i [, j]])]}

Given a list where all elements are strings or numbers,  
returns the string

@T{list[i]..sep..list[i+1] @Cdots sep..list[j]}.

The default value for @id{sep} is the empty string,

the default for @id{i} is 1,

and the default for @id{j} is @T{#list}.

If @id{i} is greater than @id{j}, returns the empty string.

}

@LibEntry{table.insert (list, [pos,] value)}

Inserts element @id{value} at position @id{pos} in @id{list},  
shifting up the elements

@T{list[pos], list[pos+1], @Cdots, list[#list]}.

The default value for @id{pos} is @T{#list+1},

so that a call @T{table.insert(t,x)} inserts @id{x} at the end  
of the list @id{t}.

}

@LibEntry{table.move (a1, f, e, t [,a2])}

Moves elements from the table @id{a1} to the table @id{a2},

performing the equivalent to the following

multiple assignment:

`@T{a2[t],@Cdots = a1[f],@Cdots,a1[e]}`.

The default for `@id{a2}` is `@id{a1}`.

The destination range can overlap with the source range.

The number of elements to be moved must fit in a Lua integer.

Returns the destination table `@id{a2}`.

}

`@LibEntry{table.pack (@Cdots)|`

Returns

a new table with all arguments stored into keys 1, 2, etc.

and with a field `@St{n}` with the total number of arguments.

Note that the resulting table may not be a sequence,

if some arguments are `@nil`.

}

`@LibEntry{table.remove (list [, pos])|`

Removes from `@id{list}` the element at position `@id{pos}`,  
returning the value of the removed element.

When `@id{pos}` is an integer between 1 and `@T{#list}`,

it shifts down the elements

`@T{list[pos+1], list[pos+2], @Cdots, list[#list]}`

and erases element `@T{list[#list]}`;

The index `@id{pos}` can also be 0 when `@T{#list}` is 0,

or `@T{#list + 1}`.

The default value for `@id{pos}` is `@T{#list}`,

so that a call `@T{table.remove(l)}` removes the last element  
of the list `@id{l}`.

}

`@LibEntry{table.sort (list [, comp])|`

Sorts the list elements in a given order, *@emph{in-place}*,  
from `@T{list[1]}` to `@T{list[#list]}`.

If `@id{comp}` is given,

then it must be a function that receives two list elements

and returns true when the first element must come

before the second in

the final order,

so that, after the sort,

`@T{i <= j}` implies `@T{not comp(list[j],list[i])}`.  
If `@id{comp}` is not given,  
then the standard Lua operator `@T{<}` is used instead.

The `@id{comp}` function must define a consistent order;  
more formally, the function must define a strict weak order.  
(A weak order is similar to a total order,  
but it can equate different elements for comparison purposes.)

The sort algorithm is not stable:  
Different elements considered equal by the given order  
may have their relative positions changed by the sort.

}

`@LibEntry{table.unpack (list [, i [, j]])}`

Returns the elements from the given list.  
This function is equivalent to  
`@verbatim{`  
return list[i], list[i+1], @Cdots, list[j]  
`}`  
By default, `@id{i}` `@N{is 1}` and `@id{j}` is `@T{#list}`.

}

}

`@sect2{mathlib| @title{Mathematical Functions}}`

This library provides basic mathematical functions.  
It provides all its functions and constants inside the table `@defid{math}`.  
Functions with the annotation `@St{integer/float}`  
give  
integer results for integer arguments  
and float results for non-integer arguments.  
The rounding functions  
`@Lid{math.ceil}`, `@Lid{math.floor}`, and `@Lid{math.modf}`  
return an integer when the result fits in the range of an integer,  
or a float otherwise.

`@LibEntry{math.abs (x)|`

Returns the maximum value between `@id{x}` and `@id{-x}`. (integer/float)

}

`@LibEntry{math.acos (x)|`



Returns the arc cosine of  $\text{@id}\{x\}$  (in radians).

}

`@LibEntry{math.asin (x)}`

Returns the arc sine of  $\text{@id}\{x\}$  (in radians).

}

`@LibEntry{math.atan (y [, x])}`

`@index{atan}` `@index{atan2}`

Returns the arc tangent of  $\text{@T}\{y/x\}$  (in radians), using the signs of both arguments to find the quadrant of the result.

It also handles correctly the case of  $\text{@id}\{x\}$  being zero.

The default value for  $\text{@id}\{x\}$  is 1, so that the call `@T{math.atan(y)}` returns the arc tangent of  $\text{@id}\{y\}$ .

}

`@LibEntry{math.ceil (x)}`

Returns the smallest integral value greater than or equal to  $\text{@id}\{x\}$ .

}

`@LibEntry{math.cos (x)}`

Returns the cosine of  $\text{@id}\{x\}$  (assumed to be in radians).

}

`@LibEntry{math.deg (x)}`

Converts the angle  $\text{@id}\{x\}$  from radians to degrees.

}

`@LibEntry{math.exp (x)}`

Returns the value  $\text{@M}\{e^{\text{@sp}\{x\}}\}$  (where  $\text{@id}\{e\}$  is the base of natural logarithms).

}

@LibEntry{math.floor (x)|

Returns the largest integral value less than or equal to @id{x}.

}

@LibEntry{math.fmod (x, y)|

Returns the remainder of the division of @id{x} by @id{y}  
that rounds the quotient towards zero. (integer/float)

}

@LibEntry{math.huge|

The float value @idx{HUGE\_VAL},  
a value greater than any other numeric value.

}

@LibEntry{math.log (x [, base])|

Returns the logarithm of @id{x} in the given base.  
The default for @id{base} is @M{e}  
(so that the function returns the natural logarithm of @id{x}).

}

@LibEntry{math.max (x, @Cdots)|

Returns the argument with the maximum value,  
according to the Lua operator @T{<}.

}

@LibEntry{math.maxinteger|

An integer with the maximum value for an integer.

}

@LibEntry{math.min (x, @Cdots)|

Returns  
the argument with the minimum value,  
according to the Lua operator @T{<}.

}

@LibEntry{math.mininteger|

An integer with the minimum value for an integer.

}

@LibEntry{math.modf (x)|

Returns the integral part of @id{x} and the fractional part of @id{x}.

Its second result is always a float.

}

@LibEntry{math.pi|

The value of @M{@pi}.

}

@LibEntry{math.rad (x)|

Converts the angle @id{x} from degrees to radians.

}

@LibEntry{math.random ([m [, n]])|

When called without arguments,  
returns a pseudo-random float with uniform distribution  
in the range @C{() @M{[0,1)}. @C{[]}

When called with two integers @id{m} and @id{n},  
@id{math.random} returns a pseudo-random integer  
with uniform distribution in the range @M{[m, n]}.

The call @T{math.random(n)}, for a positive @id{n},  
is equivalent to @T{math.random(1,n)}.

The call @T{math.random(0)} produces an integer with  
all bits (pseudo)random.

This function uses the @idx{xoshiro256\*\*} algorithm to produce  
pseudo-random 64-bit integers,  
which

are the results of calls with @N{argument 0}.

Other results (ranges and floats)

are unbiased extracted from these integers.

Lua initializes its pseudo-random generator with the equivalent of

a call to `@Lid{math.randomseed}` with no arguments,  
so that `@id{math.random}` should generate  
different sequences of results each time the program runs.

}

`@LibEntry{math.randomseed ([x [, y]])|`

When called with at least one argument,  
the integer parameters `@id{x}` and `@id{y}` are  
joined into a 128-bit `@emphx{seed}` that  
is used to reinitialize the pseudo-random generator;  
equal seeds produce equal sequences of numbers.  
The default for `@id{y}` is zero.

When called with no arguments,  
Lua generates a seed with  
a weak attempt for randomness.

This function returns the two seed components  
that were effectively used,  
so that setting them again repeats the sequence.

To ensure a required level of randomness to the initial state  
(or contrarily, to have a deterministic sequence,  
for instance when debugging  
a program),  
you should call `@Lid{math.randomseed}` with explicit arguments.

}

`@LibEntry{math.sin (x)|`

Returns the sine of `@id{x}` (assumed to be in radians).

}

`@LibEntry{math.sqrt (x)|`

Returns the square root of `@id{x}`.  
(You can also use the expression `@T{x^0.5}` to compute this value.)

}

`@LibEntry{math.tan (x)|`

Returns the tangent of `@id{x}` (assumed to be in radians).

}

@LibEntry{math.tointeger (x)|

If the value @id{x} is convertible to an integer,  
returns that integer.

Otherwise, returns @fail.

}

@LibEntry{math.type (x)|

Returns @St{integer} if @id{x} is an integer,

@St{float} if it is a float,

or @fail if @id{x} is not a number.

}

@LibEntry{math.ult (m, n)|

Returns a boolean,

@true if and only if integer @id{m} is below integer @id{n} when  
they are compared as @x{unsigned integers}.

}

}

@sect2{iolib| @title{Input and Output Facilities}}

The I/O library provides two different styles for file manipulation.

The first one uses implicit file handles;

that is, there

are operations to set a default input file and a

default output file,

and all input/output operations are done over these default files.

The second style uses explicit file handles.

When using implicit file handles,

all operations are supplied by table @defid{io}.

When using explicit file handles,

the operation @Lid{io.open} returns a file handle

and then all operations are supplied as methods of the file handle.

The metatable for file handles provides metamethods

for @idx{\_\_gc} and @idx{\_\_close} that try

to close the file when called.

The table @id{io} also provides  
three predefined file handles with their usual meanings from C:  
@defid{io.stdin}, @defid{io.stdout}, and @defid{io.stderr}.  
The I/O library never closes these files.

Unless otherwise stated,  
all I/O functions return @fail on failure,  
plus an error message as a second result and  
a system-dependent error code as a third result,  
and some non-false value on success.  
On non-POSIX systems,  
the computation of the error message  
and error code  
in case of errors  
may be not @x{thread safe},  
because they rely on the global C variable @id{errno}.

@LibEntry{io.close ([file])|

Equivalent to @T{file:close()}.  
Without a @id{file}, closes the default output file.

}

@LibEntry{io.flush ()|

Equivalent to @T{io.output():flush()}.

}

@LibEntry{io.input ([file])|

When called with a file name, it opens the named file (in text mode),  
and sets its handle as the default input file.  
When called with a file handle,  
it simply sets this file handle as the default input file.  
When called without arguments,  
it returns the current default input file.

In case of errors this function raises the error,  
instead of returning an error code.

}

@LibEntry{io.lines ([filename, @Cdots])|

Opens the given file name in read mode  
and returns an iterator function that

works like `@T{file:lines(@Cdots)}` over the opened file.

When the iterator function fails to read any value,

it automatically closes the file.

Besides the iterator function,

`@id{io.lines}`

returns three other values:

two `@nil` values as placeholders,

plus the created file handle.

Therefore, when used in a generic `@Rw{for}` loop,

the file is closed also if the loop is interrupted by an

error or a `@Rw{break}`.

The call `@T{io.lines()}` (with no file name) is equivalent

to `@T{io.input():lines("")}`;

that is, it iterates over the lines of the default input file.

In this case, the iterator does not close the file when the loop ends.

In case of errors opening the file,

this function raises the error,

instead of returning an error code.

}

`@LibEntry{io.open (filename [, mode])|`

This function opens a file,

in the mode specified in the string `@id{mode}`.

In case of success,

it returns a new file handle.

The `@id{mode}` string can be any of the following:

`@description{`

`@item{@St{r}| read mode (the default);}`

`@item{@St{w}| write mode;}`

`@item{@St{a}| append mode;}`

`@item{@St{r+}| update mode, all previous data is preserved;}`

`@item{@St{w+}| update mode, all previous data is  
erased;}`

`@item{@St{a+}| append update mode, previous data is preserved,  
writing is only allowed at the end of file.}`

}

The `@id{mode}` string can also have a `@Char{b}` at the end,

which is needed in some systems to open the file in binary mode.

}

`@LibEntry{io.output ([file])|`

Similar to @Lid{io.input}, but operates over the default output file.

```
}
```

```
@LibEntry{io.popen (prog [, mode])|
```

This function is system dependent and is not available on all platforms.

Starts the program @id{prog} in a separated process and returns a file handle that you can use to read data from this program (if @id{mode} is @T{"r"}, the default) or to write data to this program (if @id{mode} is @T{"w"}).

```
}
```

```
@LibEntry{io.read (@Cdots)|
```

Equivalent to @T{io.input():read(@Cdots)}.

```
}
```

```
@LibEntry{io.tmpfile ()|
```

In case of success,  
returns a handle for a temporary file.  
This file is opened in update mode  
and it is automatically removed when the program ends.

```
}
```

```
@LibEntry{io.type (obj)|
```

Checks whether @id{obj}  
is a valid file handle.  
Returns the string @T{"file"} if @id{obj} is an open file handle,  
@T{"closed file"} if @id{obj} is a closed file handle,  
or @fail if @id{obj} is not a file handle.

```
}
```

```
@LibEntry{io.write (@Cdots)|
```

Equivalent to @T{io.output():write(@Cdots)}.

```
}
```



```
@LibEntry{file:close ()|
```

Closes @id{file}.

Note that files are automatically closed when their handles are garbage collected, but that takes an unpredictable amount of time to happen.

When closing a file handle created with @Lid{io.popen}, @Lid{file:close} returns the same values returned by @Lid{os.execute}.

```
}
```

```
@LibEntry{file:flush ()|
```

Saves any written data to @id{file}.

```
}
```

```
@LibEntry{file:lines (@Cdots)|
```

Returns an iterator function that, each time it is called, reads the file according to the given formats. When no format is given, uses @St{1} as a default.

As an example, the construction

```
@verbatim{  
for c in file:lines(1) do @rep{body} end  
}
```

will iterate over all characters of the file, starting at the current position.

Unlike @Lid{io.lines}, this function does not close the file when the loop ends.

```
}
```

```
@LibEntry{file:read (@Cdots)|
```

Reads the file @id{file}, according to the given formats, which specify what to read.

For each format,

the function returns a string or a number with the characters read, or @fail if it cannot read data with the specified format.

(In this latter case,

the function does not read subsequent formats.)

When called without arguments,  
it uses a default format that reads the next line  
(see below).

The available formats are  
`@description{`

`@item{@St{n}|`  
reads a numeral and returns it as a float or an integer,  
following the lexical conventions of Lua.  
(The numeral may have leading whitespaces and a sign.)  
This format always reads the longest input sequence that  
is a valid prefix for a numeral;  
if that prefix does not form a valid numeral  
(e.g., an empty string, `@St{0x}`, or `@St{3.4e-}`)  
or it is too long (more than 200 characters),  
it is discarded and the format returns `@fail`.  
`}`

`@item{@St{a}|`  
reads  
the whole file, starting at the current position.  
On end of file, it returns the empty string;  
this format never fails.  
`}`

`@item{@St{1}|`  
reads the next line skipping the end of line,  
returning `@fail` on end of file.  
This is the default format.  
`}`

`@item{@St{L}|`  
reads the next line keeping the end-of-line character (if present),  
returning `@fail` on end of file.  
`}`

`@item{@emph{number}|`  
reads a string with up to this number of bytes,  
returning `@fail` on end of file.  
If `@id{number}` is zero,  
it reads nothing and returns an empty string,  
or `@fail` on end of file.  
`}`

`}`

The formats `@St{1}` and `@St{L}` should be used only for text files.

}

@LibEntry{file:seek ([whence [, offset]])|

Sets and gets the file position,  
measured from the beginning of the file,  
to the position given by @id{offset} plus a base  
specified by the string @id{whence}, as follows:

@description{  
@item{@St{set}| base is position 0 (beginning of the file);}  
@item{@St{cur}| base is current position;}  
@item{@St{end}|  
base is end of file;}  
}

In case of success, @id{seek} returns the final file position,  
measured in bytes from the beginning of the file.

If @id{seek} fails, it returns @fail,  
plus a string describing the error.

The default value for @id{whence} is @T{"cur"},  
and for @id{offset} is 0.

Therefore, the call @T{file:seek()} returns the current  
file position, without changing it;  
the call @T{file:seek("set")} sets the position to the  
beginning of the file (and returns 0);  
and the call @T{file:seek("end")} sets the position to the  
end of the file, and returns its size.

}

@LibEntry{file:setvbuf (mode [, size])|

Sets the buffering mode for a file.

There are three available modes:

@description{  
@item{@St{no}| no buffering.}  
@item{@St{full}| full buffering.}  
@item{@St{line}| line buffering.}  
}

For the last two cases,  
@id{size} is a hint for the size of the buffer, in bytes.  
The default is an appropriate size.

The specific behavior of each mode is non portable;  
check the underlying  
@ANSI{setvbuf} in your platform for  
more details.

```
}
```

```
@LibEntry{file:write (@Cdots)|
```

Writes the value of each of its arguments to @id{file}.

The arguments must be strings or numbers.

In case of success, this function returns @id{file}.

```
}
```

```
}
```

```
@sect2{oslib| @title{Operating System Facilities}
```

This library is implemented through table @defid{os}.

```
@LibEntry{os.clock ()|
```

Returns an approximation of the amount in seconds of CPU time used by the program,

as returned by the underlying @ANSI{clock}.

```
}
```

```
@LibEntry{os.date ([format [, time]])|
```

Returns a string or a table containing date and time, formatted according to the given string @id{format}.

If the @id{time} argument is present,  
this is the time to be formatted  
(see the @Lid{os.time} function for a description of this value).  
Otherwise, @id{date} formats the current time.

If @id{format} starts with @Char{!},  
then the date is formatted in Coordinated Universal Time.  
After this optional character,  
if @id{format} is the string @St{\*t},  
then

@id{date} returns a table with the following fields:  
@id{year}, @id{month} (1@En{ }12), @id{day} (1@En{ }31),  
@id{hour} (0@En{ }23), @id{min} (0@En{ }59),  
@id{sec} (0@En{ }61, due to leap seconds),  
@id{yday} (weekday, 1@En{ }7, Sunday @N{is 1}),  
@id{yday} (day of the year, 1@En{ }366),

and `@id{isdst}` (daylight saving flag, a boolean).

This last field may be absent  
if the information is not available.

If `@id{format}` is not `@St{*t}`,  
then `@id{date}` returns the date as a string,  
formatted according to the same rules as the `@ANSI{strftime}`.

If `@id{format}` is absent, it defaults to `@St{%c}`,  
which gives a human-readable date and time representation  
using the current locale.

On non-POSIX systems,  
this function may be not `@x{thread safe}`  
because of its reliance on `@Cid{gmtime}` and `@Cid{localtime}`.

}

`@LibEntry{os.difftime (t2, t1)|`

Returns the difference, in seconds,  
from time `@id{t1}` to time `@id{t2}`  
(where the times are values returned by `@Lid{os.time}`).  
In `@x{POSIX}`, `@x{Windows}`, and  
some other systems,  
this value is exactly `@id{t2}@M{-}@id{t1}`.

}

`@LibEntry{os.execute ([command])|`

This function is equivalent to the `@ANSI{system}`.  
It passes `@id{command}` to be executed by an operating system shell.  
Its first result is `@true`  
if the command terminated successfully,  
or `@fail` otherwise.  
After this first result  
the function returns a string plus a number,  
as follows:  
`@description{`

`@item{@St{exit}|`  
the command terminated normally;  
the following number is the exit status of the command.  
}

`@item{@St{signal}|`  
the command was terminated by a signal;

the following number is the signal that terminated the command.

}

}

When called without a @id{command},

@id{os.execute} returns a boolean that is true if a shell is available.

}

@LibEntry{os.exit ([code [, close]])|

Calls the @ANSI{exit} to terminate the host program.

If @id{code} is @true,

the returned status is @idx{EXIT\_SUCCESS};

if @id{code} is @false,

the returned status is @idx{EXIT\_FAILURE};

if @id{code} is a number,

the

returned status is this number.

The default value for @id{code} is @true.

If the optional second argument @id{close} is true,

the function closes the Lua state before exiting @seeF{lua\_close}.

}

@LibEntry{os.getenv (varname)|

Returns the value of the process environment variable @id{varname}

or @fail if the variable is not defined.

}

@LibEntry{os.remove (filename)|

Deletes the file (or empty directory, on @x{POSIX} systems)

with the given name.

If this function fails, it returns @fail

plus a string describing the error and the error code.

Otherwise, it returns true.

}

@LibEntry{os.rename (oldname, newname)|

Renames the file or directory named @id{oldname} to @id{newname}.

If this function fails, it returns @fail,

plus a string describing the error and the error code.

Otherwise, it returns true.

}

@LibEntry{os.setlocale (locale [, category])|

Sets the current locale of the program.

@id{locale} is a system-dependent string specifying a locale;

@id{category} is an optional

string describing which category to change:

@T{"all"}, @T{"collate"}, @T{"ctype"},

@T{"monetary"}, @T{"numeric"}, or @T{"time"};

the default category is @T{"all"}.

The function returns the name of the new locale,

or @fail if the request cannot be honored.

If @id{locale} is the empty string,

the current locale is set to an implementation-defined native locale.

If @id{locale} is the string @St{C},

the current locale is set to the standard C locale.

When called with @nil as the first argument,

this function only returns the name of the current locale

for the given category.

This function may be not @x{thread safe}

because of its reliance on @CId{setlocale}.

}

@LibEntry{os.time ([table])|

Returns the current time when called without arguments,

or a time representing the local date and time specified by the given table.

This table must have fields @id{year}, @id{month}, and @id{day},

and may have fields

@id{hour} (default is 12),

@id{min} (default is 0),

@id{sec} (default is 0),

and

@id{isdst} (default is @nil).

Other fields are ignored.

For a description of these fields, see the @Lid{os.date} function.

When the function is called,

the values in these fields do not need to be inside their valid ranges.

For instance, if @id{sec} is -10,

it means 10 seconds before the time specified by the other fields;  
if @id{hour} is 1000,  
it means 1000 hours after the time specified by the other fields.

The returned value is a number, whose meaning depends on your system.  
In @x{POSIX}, @x{Windows}, and some other systems,  
this number counts the number  
of seconds since some given start time (the @Q{epoch}).  
In other systems, the meaning is not specified,  
and the number returned by @id{time} can be used only as an argument to  
@Lid{os.date} and @Lid{os.difftime}.

When called with a table,  
@id{os.time} also normalizes all the fields  
documented in the @Lid{os.date} function,  
so that they represent the same time as before the call  
but with values inside their valid ranges.

}

@LibEntry{os.tmpname  
()|

Returns a string with a file name that can  
be used for a temporary file.  
The file must be explicitly opened before its use  
and explicitly removed when no longer needed.

In @x{POSIX} systems,  
this function also creates a file with that name,  
to avoid security risks.  
(Someone else might create the file with wrong permissions  
in the time between getting the name and creating the file.)  
You still have to open the file to use it  
and to remove it (even if you do not use it).

When possible,  
you may prefer to use @Lid{io.tmpfile},  
which automatically removes the file when the program ends.

}

}

@sect2{debuglib| @title{The Debug Library}}

This library provides  
the functionality of the @link{debugI|debug interface} to Lua programs.



You should exert care when using this library.  
Several of its functions  
violate basic assumptions about Lua code  
(e.g., that variables local to a function  
cannot be accessed from outside;  
that userdata metatables cannot be changed by Lua  
code;  
that Lua programs do not crash)  
and therefore can compromise otherwise secure code.  
Moreover, some functions in this library may be slow.

All functions in this library are provided  
inside the `@defid{debug}` table.  
All functions that operate over a thread  
have an optional first argument which is the  
thread to operate over.  
The default is always the current thread.

`@LibEntry{debug.debug ()|`

Enters an interactive mode with the user,  
running each string that the user enters.  
Using simple commands and other debug facilities,  
the user can inspect global and local variables,  
change their values, evaluate expressions, and so on.  
A line containing only the word `@id{cont}` finishes this function,  
so that the caller continues its execution.

Note that commands for `@id{debug.debug}` are not lexically nested  
within any function and so have no direct access to local variables.

`}`

`@LibEntry{debug.gethook ([thread])|`

Returns the current hook settings of the thread, as three values:  
the current  
hook function, the current hook mask,  
and the current hook count,  
as set by the `@Lid{debug.sethook}` function.

Returns `@fail` if there is no active hook.

`}`

`@LibEntry{debug.getinfo ([thread,] f [, what])|`

Returns a table with information about a function.

You can give the function directly

or you can give a number as the value of `@id{f}`,

which means the function running at level `@id{f}` of the call stack of the given thread:

`@N{level 0}` is the current function (`@id{getinfo}` itself);

`@N{level 1}` is the function that called `@id{getinfo}`

(except for tail calls, which do not count in the stack);

and so on.

If `@id{f}` is a number greater than the number of active functions, then `@id{getinfo}` returns `@fail`.

The returned table can contain all the fields returned by `@Lid{lua_getinfo}`, with the string `@id{what}` describing which fields to fill in.

The default for `@id{what}` is to get all information available, except the table of valid lines.

The option `@Char{f}`

adds a field named `@id{func}` with the function itself.

The option `@Char{L}` adds a field named `@id{activelines}` with the table of valid lines, provided the function is a Lua function.

If the function has no debug information, the table is empty.

For instance, the expression `@T{debug.getinfo(1,"n").name}` returns a name for the current function, if a reasonable name can be found, and the expression `@T{debug.getinfo(print)}` returns a table with all available information about the `@Lid{print}` function.

}

`@LibEntry{debug.getlocal ([thread,] f, local)}`

This function returns the name and the value of the local variable with index `@id{local}` of the function at level `@id{f}` of the stack. This function accesses not only explicit local variables, but also parameters and temporary values.

The first parameter or local variable has `@N{index 1}`, and so on, following the order that they are declared in the code, counting only the variables that are active in the current scope of the function.

Compile-time constants may not appear in this listing,

if they were optimized away by the compiler.

Negative indices refer to vararg arguments;  
@num{-1} is the first vararg argument.  
The function returns @fail  
if there is no variable with the given index,  
and raises an error when called with a level out of range.  
(You can call @Lid{debug.getinfo} to check whether the level is valid.)

Variable names starting with @Char{() (open parenthesis) @C{}}  
represent variables with no known names  
(internal variables such as loop control variables,  
and variables from chunks saved without debug information).

The parameter @id{f} may also be a function.  
In that case, @id{getlocal} returns only the name of function parameters.

}

@LibEntry{debug.getmetatable (value)|

Returns the metatable of the given @id{value}  
or @nil if it does not have a metatable.

}

@LibEntry{debug.getregistry ()|

Returns the registry table @see{registry}.

}

@LibEntry{debug.getupvalue (f, up)|

This function returns the name and the value of the upvalue  
with

index @id{up} of the function @id{f}.

The function returns @fail  
if there is no upvalue with the given index.

(For Lua functions,  
upvalues are the external local variables that the function uses,  
and that are consequently included in its closure.)

For @N{C functions}, this function uses the empty string @T{""}  
as a name for all upvalues.

Variable name @Char{?} (interrogation mark)  
represents variables with no known names  
(variables from chunks saved without debug information).

```
}
```

```
@LibEntry{debug.getuservalue (u, n)|
```

Returns the @id{n}-th user value associated to the userdata @id{u} plus a boolean, @false if the userdata does not have that value.

```
}
```

```
@LibEntry{debug.sethook ([thread,] hook, mask [, count])|
```

Sets the given function as the debug hook.

The string @id{mask} and the number @id{count} describe when the hook will be called.

The string mask may have any combination of the following characters, with the given meaning:

```
@description{
```

```
@item{@Char{c}| the hook is called every  
time Lua calls a function;}
```

```
@item{@Char{r}| the hook is called every time Lua returns from a function;}
```

```
@item{@Char{l}| the hook is called every time Lua enters a new line of code.}
```

```
}
```

Moreover,

with a @id{count} different from zero,

the hook is called also after every @id{count} instructions.

When called without arguments,

@Lib{debug.sethook} turns off the hook.

When the hook is called, its first parameter is a string describing the event that has triggered its call:

@T{"call"}, @T{"tail call"}, @T{"return"},

@T{"line"}, and @T{"count"}.

For line events,

the hook also gets the new line number as its second parameter.

Inside a hook,

you can call @id{getinfo} with @N{level 2} to get more information about the running function.

(@N{Level 0} is the @id{getinfo} function,

and @N{level 1} is the hook function.)

```
}
```

```
@LibEntry{debug.setlocal ([thread,] level, local, value)|
```

This function assigns the value @id{value} to the local variable

with index `@id{local}` of the function at level `@id{level}` of the stack.  
The function returns `@fail` if there is no local variable with the given index, and raises an error when called with a `@id{level}` out of range. (You can call `@id{getinfo}` to check whether the level is valid.) Otherwise, it returns the name of the local variable.

See `@Lid{debug.getlocal}` for more information about variable indices and names.

}

`@LibEntry{debug.setmetatable (value, table)|`

Sets the metatable for the given `@id{value}` to the given `@id{table}` (which can be `@nil`).  
Returns `@id{value}`.

}

`@LibEntry{debug.setupvalue (f, up, value)|`

This function assigns the value `@id{value}` to the upvalue with index `@id{up}` of the function `@id{f}`.  
The function returns `@fail` if there is no upvalue with the given index.  
Otherwise, it returns the name of the upvalue.

See `@Lid{debug.getupvalue}` for more information about upvalues.

}

`@LibEntry{debug.setuservalue (udata, value, n)|`

Sets the given `@id{value}` as the `@id{n}`-th user value associated to the given `@id{udata}`.  
`@id{udata}` must be a full userdata.

Returns `@id{udata}`,  
or `@fail` if the userdata does not have that value.

}

`@LibEntry{debug.traceback ([thread,] [message [, level]])|`

If `@id{message}` is present but is neither a string nor `@nil`,

this function returns @id{message} without further processing.

Otherwise,

it returns a string with a traceback of the call stack.

The optional @id{message} string is appended at the beginning of the traceback.

An optional @id{level} number tells at which level to start the traceback

(default is 1, the function calling @id{traceback}).

}

@LibEntry{debug.upvalueid (f, n)|

Returns a unique identifier (as a light userdata)

for the upvalue numbered @id{n}

from the given function.

These unique identifiers allow a program to check whether different closures share upvalues.

Lua closures that share an upvalue

(that is, that access a same external local variable)

will return identical ids for those upvalue indices.

}

@LibEntry{debug.upvaluejoin (f1, n1, f2, n2)|

Make

the @id{n1}-th upvalue of the Lua closure @id{f1}

refer to the @id{n2}-th upvalue of the Lua closure @id{f2}.

}

}

}

@C{-----}

@sect1{lua-sa| @title{Lua Standalone}}

Although Lua has been designed as an extension language,

to be embedded in a host @N{C program},

it is also frequently used as a standalone language.

An interpreter for Lua as a standalone language,

called simply @id{lua},

is provided with the standard distribution.

The @x{standalone interpreter} includes

all standard libraries.

Its usage is:

```
@verbatim{
lua [options] [script [args]]
}
```

The options are:

```
@description{
@item{@T{-e @rep{stat}}| execute string @rep{stat};}
@item{@T{-i}| enter interactive mode after running @rep{script};}
@item{@T{-l @rep{mod}}| @Q{require} @rep{mod} and assign the
result to global @rep{mod};}
@item{@T{-l @rep{g=mod}}| @Q{require} @rep{mod} and assign the
result to global @rep{g};}
@item{@T{-v}| print version information;}
@item{@T{-E}|
ignore environment variables;}
@item{@T{-W}| turn warnings on;}
@item{@T{--}| stop handling options;}
@item{@T{-}| execute @id{stdin} as a file and stop handling options.}
}
```

(The form @T{-l @rep{g=mod}} was introduced in @N{release 5.4.4}.)

After handling its options, @id{lua} runs the given @emph{script}.

When called without arguments,

@id{lua} behaves as @T{lua -v -i}

when the standard input (@id{stdin}) is a terminal,

and as @T{lua -} otherwise.

When called without the option @T{-E},

the interpreter checks for an environment variable @defid{LUA\_INIT\_5\_4}  
(or @defid{LUA\_INIT} if the versioned name is not defined)

before running any argument.

If the variable content has the format @T{@At@rep{filename}},

then @id{lua} executes the file.

Otherwise, @id{lua} executes the string itself.

When called with the option @T{-E},

Lua does not consult any environment variables.

In particular,

the values of @Lid{package.path} and @Lid{package.cpath}  
are set with the default

paths defined in @id{luaconf.h}.

To signal to the libraries that this option is on,

the stand-alone interpreter sets the field

@idx{"LUA\_NOENV"} in the registry to a true value.

Other libraries may consult this field for the same purpose.

The options @T{-e}, @T{-l}, and @T{-W} are handled in

the order they appear.

For instance, an invocation like

```
@verbatim{
$ lua -e 'a=1' -llib1 script.lua
}
```

will first set `@id{a}` to 1, then require the library `@id{lib1}`,

and finally run the file `@id{script.lua}` with no arguments.

(Here `@T{$}` is the shell prompt. Your prompt may be different.)

Before running any code,

`@id{lua}` collects all command-line arguments

in a global table called `@id{arg}`.

The script name goes to index 0,

the first argument after the script name goes to index 1,

and so on.

Any arguments before the script name

(that is, the interpreter name plus its options)

go to negative indices.

For instance, in the call

```
@verbatim{
$ lua -la b.lua t1 t2
}
```

the table is like this:

```
@verbatim{
arg
= { [-2] = "lua", [-1] = "-la",
    [0] = "b.lua",
    [1] = "t1", [2] = "t2" }
}
```

If there is no script in the call,

the interpreter name goes to index 0,

followed by the other arguments.

For instance, the call

```
@verbatim{
$ lua -e "print(arg[1])"
}
```

will print `@St{-e}`.

If there is a script,

the script is called with arguments

`@T{arg[1]}`, `@Cdots`, `@T{arg[#arg]}`.

Like all chunks in Lua,

the script is compiled as a variadic function.

In interactive mode,

Lua repeatedly prompts and waits for a line.

After reading a line,

Lua first try to interpret the line as an expression.

If it succeeds, it prints its value.



Otherwise, it interprets the line as a statement.

If you write an incomplete statement,  
the interpreter waits for its completion  
by issuing a different prompt.

If the global variable `@defid{_PROMPT}` contains a string,  
then its value is used as the prompt.

Similarly, if the global variable `@defid{_PROMPT2}` contains a string,  
its value is used as the secondary prompt  
(issued  
during incomplete statements).

In case of unprotected errors in the script,  
the interpreter reports the error to the standard error stream.

If the error object is not a string but  
has a metamethod `@idx{__tostring}`,  
the interpreter calls this metamethod to produce the final message.  
Otherwise, the interpreter converts the error object to a string  
and adds a stack traceback to it.

When warnings are on,  
they are simply printed in the standard error output.

When finishing normally,  
the interpreter closes its main Lua state  
`@seeF{lua_close}`.

The script can avoid this step by  
calling `@Lid{os.exit}` to terminate.

To allow the use of Lua as a  
script interpreter in Unix systems,  
Lua skips the first line of a file chunk if it starts with `@T{#}`.  
Therefore, Lua scripts can be made into executable programs  
by using `@T{chmod +x}` and `@N{the @T{#!}}` form,  
as in

```
@verbatim{
#!/usr/local/bin/lua
}
```

Of course,  
the location of the Lua interpreter may be different in your machine.  
If `@id{lua}` is in your

`@id{PATH}`,  
then

```
@verbatim{
#!/usr/bin/env lua
}
```

is a more portable solution.

```
}
```

@sect1{incompat| @title{Incompatibilities with the Previous Version}

@simplesect{

Here we list the incompatibilities that you may find when moving a program from @N{Lua 5.3} to @N{Lua 5.4}.

You can avoid some incompatibilities by compiling Lua with appropriate options (see file @id{luaconf.h}).

However,

all these compatibility options will be removed in the future.

More often than not,

compatibility issues arise when these compatibility options are removed.

So, whenever you have the chance,

you should try to test your code with a version of Lua compiled with all compatibility options turned off.

That will ease transitions to newer versions of Lua.

Lua versions can always change the C API in ways that

do not imply source-code changes in a program,

such as the numeric values for constants

or the implementation of functions as macros.

Therefore,

you should never assume that binaries are compatible between different

Lua versions.

Always recompile clients of the Lua API when using a new version.

Similarly, Lua versions can always change the internal representation

of precompiled chunks;

precompiled chunks are not compatible between different Lua versions.

The standard paths in the official distribution may

change between versions.

}

@sect2{@title{Incompatibilities in the Language}

@itemize{

@item{

The coercion of strings to numbers in

arithmetic and bitwise operations

has been removed from the core language.

The string library does a similar job  
for arithmetic (but not for bitwise) operations  
using the string metamethods.  
However, unlike in previous versions,  
the new implementation preserves the implicit type of the numeral  
in the string.  
For instance, the result of `@T{"1" + "2"}` now is an integer,  
not a float.  
}

@item{  
Literal decimal integer constants that overflow are read as floats,  
instead of wrapping around.  
You can use hexadecimal notation for such constants if you  
want  
the old behavior  
(reading them as integers with wrap around).  
}

@item{  
The use of the `@idx{__lt}` metamethod to emulate `@idx{__le}`  
has been removed.  
When needed, this metamethod must be explicitly defined.  
}

@item{  
The semantics of the numerical `@Rw{for}` loop  
over integers changed in some details.  
In particular, the control variable never wraps around.  
}

@item{  
A label for a `@Rw{goto}` cannot be declared where a label with the same  
name is visible, even if this other label is declared in an enclosing  
block.  
}

@item{  
When finalizing an object,  
Lua does not ignore `@idx{__gc}` metamethods that are not functions.  
Any value will be called, if present.  
(Non-callable values will generate a warning,  
like any other error when calling a finalizer.)  
}  
  
}  
  
}

@sect2{@title{Incompatibilities in the Libraries}

@itemize{

@item{

The function @Lid{print} does not call @Lid{tostring} to format its arguments;

instead, it has this functionality hardwired.

You should use @idx{\_\_tostring} to modify how values are printed.

}

@item{

The pseudo-random number generator used by the function @Lid{math.random} now starts with a somewhat random seed.

Moreover, it uses a different algorithm.

}

@item{

By default, the decoding functions in the @Lid{utf8} library do not accept surrogates as valid code points.

An extra parameter in these functions makes them more permissive.

}

@item{

The options @St{setpause} and @St{setstepmul} of the function @Lid{collectgarbage} are deprecated.

You should use the new option @St{incremental} to set them.

}

@item{

The function @Lid{io.lines} now returns four values, instead of just one.

That can be a problem when it is used as the sole argument to another function that has optional parameters, such as in @T{load(io.lines(filename, "L"))}.

To fix that issue, you can wrap the call into parentheses, to adjust its number of results to one.

}

}

}

@sect2{@title{Incompatibilities in the API}

@itemize{

@item{  
Full userdata now has an arbitrary number of associated  
user values.  
Therefore, the functions @id{lua\_newuserdata},  
@id{lua\_setuservalue}, and @id{lua\_getuservalue} were  
replaced by @Lid{lua\_newuserdatauv},  
@Lid{lua\_setiuservalue}, and @Lid{lua\_getiuservalue},  
which have an extra argument.

For compatibility, the old names still work as macros assuming  
one single user value.

Note, however, that userdata with zero user values  
are more efficient memory-wise.

}

@item{  
The function @Lid{lua\_resume} has an extra parameter.  
This out parameter returns the number of values on  
the top of the stack that were yielded or returned by the coroutine.  
(In previous versions,  
those values were the entire stack.)  
}

@item{  
The function @Lid{lua\_version} returns the version number,  
instead of an address of the version number.  
The Lua core should work correctly with libraries using their  
own static copies of the same core,  
so there is no need to check whether they are using the same  
address space.  
}

@item{  
The constant @id{LUA\_ERRGCM} was removed.  
Errors in  
finalizers are never propagated;  
instead, they generate a warning.  
}

@item{  
The options @idx{LUA\_GCSETPAUSE} and @idx{LUA\_GCSETSTEPMUL}  
of the function @Lid{lua\_gc} are deprecated.  
You should use the new option @id{LUA\_GCINC} to set them.  
}

}

}

}

@C{[=====]}

@sect1{BNF| @title{The Complete Syntax of Lua}

Here is the complete syntax of Lua in extended BNF.

As usual in extended BNF,

@bnfNter{A} means 0 or more @bnfNter{A}s,

and @bnfNter{[A]} means an optional @bnfNter{A}.

(For operator precedences, see @See{prec};

for a description of the terminals

@bnfNter{Name}, @bnfNter{Numeral},

and @bnfNter{LiteralString}, see @See{lexical}.)

@index{grammar}

@Produc{

@producname{chunk}@producbody{block}

@producname{block}@producbody{@bnfrep{stat} @bnfopt{retstat}}

@producname{stat}@producbody{

@bnfter{;}

@OrNL varlist @bnfter{=} explist

@OrNL functioncall

@OrNL label

@OrNL @Rw{break}

@OrNL @Rw{goto} Name

@OrNL @Rw{do} block

@Rw{end}

@OrNL @Rw{while} exp @Rw{do} block @Rw{end}

@OrNL @Rw{repeat} block @Rw{until} exp

@OrNL @Rw{if} exp @Rw{then} block

@bnfrep{@Rw{elseif} exp @Rw{then} block}

@bnfopt{@Rw{else} block} @Rw{end}

@OrNL @Rw{for} @bnfNter{Name} @bnfter{=} exp @bnfter{,} exp @bnfopt{@bnfter{,} exp}

@Rw{do} block @Rw{end}

@OrNL @Rw{for} namelist @Rw{in} explist @Rw{do} block @Rw{end}

@OrNL @Rw{function} funcname funcbody

@OrNL @Rw{local} @Rw{function} @bnfNter{Name} funcbody

@OrNL @Rw{local} attnamelist @bnfopt{@bnfter{=} explist}

}

@producname{attnamelist}@producbody{

```

@bnfNter{Name} attrib @bnfrep{@bnfter{,} @bnfNter{Name} attrib}}

@producname{attrib}@producbody{@bnfopt{@bnfter{<} @bnfNter{Name} @bnfter{>}}}

@producname{retstat}@producbody{@Rw{return}
    @bnfopt{explist} @bnfopt{@bnfter{;}}}}

@producname{label}@producbody{@bnfter{::} Name @bnfter{::}}

@producname{funcname}@producbody{@bnfNter{Name} @bnfrep{@bnfter{.} @bnfNter{Name}}

    @bnfopt{@bnfter{:} @bnfNter{Name}}}}

@producname{varlist}@producbody{var @bnfrep{@bnfter{,} var}}

@producname{var}@producbody{
    @bnfNter{Name}
    @Or prefixexp @bnfter{[] exp @bnfter{[]}}
    @Or prefixexp @bnfter{.} @bnfNter{Name}
}

@producname{namelist}@producbody{@bnfNter{Name} @bnfrep{@bnfter{,} @bnfNter{Name}}}}

@producname{explist}@producbody{exp @bnfrep{@bnfter{,} exp}}

@producname{exp}@producbody{
    @Rw{nil}
    @Or @Rw{false}
    @Or @Rw{true}
    @Or @bnfNter{Numeral}
    @Or @bnfNter{LiteralString}
    @Or @bnfter{...}
    @Or functiondef
    @OrNL prefixexp
    @Or tableconstructor
    @Or exp binop exp
    @Or unop exp
}

@producname{prefixexp}@producbody{var @Or functioncall @Or @bnfter{() exp @bnfter{)}}}

@producname{functioncall}@producbody{
    prefixexp args
    @Or prefixexp @bnfter{:} @bnfNter{Name} args
}

@producname{args}@producbody{

```

```

@bnfter{() @bnfopt{explist} @bnfter{}}
@Or tableconstructor
@Or @bnfNter{LiteralString}
}

@producname{functiondef}@producbody{@Rw{function} funcbody}

@producname{funcbody}@producbody{@bnfter{()
@bnfopt{parlist} @bnfter{)}} block @Rw{end}}

@producname{parlist}@producbody{namelist @bnfopt{@bnfter{,} @bnfter{...}}
@Or @bnfter{...}}

@producname{tableconstructor}@producbody{@bnfter{@Open} @bnfopt{fieldlist} @bnfter{@Close}}

@producname{fieldlist}@producbody{field @bnfrep{fieldsep field} @bnfopt{fieldsep}}

@producname{field}@producbody{@bnfter{[] exp @bnfter{[]} @bnfter{=} exp @Or @bnfNter{Name}
@bnfter{=} exp @Or exp}

@producname{fieldsep}@producbody{@bnfter{,} @Or @bnfter{;}}

@producname{binop}@producbody{
@bnfter{+} @Or @bnfter{-} @Or @bnfter{*} @Or @bnfter{/} @Or @bnfter{//}
@Or @bnfter{^} @Or @bnfter{%}
@OrNL
@bnfter{&} @Or @bnfter{~} @Or @bnfter{[]} @Or @bnfter{>>} @Or @bnfter{<<}
@Or @bnfter{..}
@OrNL
@bnfter{<} @Or @bnfter{<=} @Or @bnfter{>} @Or @bnfter{>=}
@Or @bnfter{==} @Or @bnfter{~=}
@OrNL
@Rw{and} @Or @Rw{or}}

@producname{unop}@producbody{@bnfter{-} @Or @Rw{not} @Or @bnfter{#} @Or
@bnfter{~}}

}

}

@C{ }=====}

}
@C{ }-----}

```

Found



in path(s):

\* /lua-5-4-7-1-tar-gz/lua-5.4.7/manual/manual.of

No license file was found, but licenses were detected in source scan.

/\*\*\*\*\*

\* Copyright (C) 1994-2024 Lua.org, PUC-Rio.

\*

\* Permission is hereby granted, free of charge, to any person obtaining

\* a copy of this software and associated documentation files (the

\* "Software"), to deal in the Software without restriction, including

\* without limitation the rights to use, copy, modify, merge, publish,

\* distribute, sublicense, and/or sell copies of the Software, and to

\* permit persons to whom the Software is furnished to do so, subject to

\* the following conditions:

\*

\* The above copyright notice and this permission notice shall be

\* included in all copies or substantial portions of the Software.

\*

\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,

\* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF

\* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

\* IN NO EVENT

SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY

\* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,

\* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE

\* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

\*\*\*\*\*/

Found in path(s):

\* /lua-5-4-7-1-tar-gz/lua-5.4.7/lua.h

No license file was found, but licenses were detected in source scan.

--[[

\*\*\*\*\*

\* Copyright (C) 1994-2016 Lua.org, PUC-Rio.

\*

\* Permission is hereby granted, free of charge, to any person obtaining

\* a copy of this software and associated documentation files (the

\* "Software"), to deal in the Software without restriction, including

\* without limitation the rights to use, copy, modify, merge, publish,

\* distribute, sublicense, and/or sell copies of the Software, and to

\* permit persons to whom the Software is furnished to do so, subject to

\* the following conditions:

\*

\* The above copyright notice and this permission notice shall be

\* included in all copies or substantial portions of the Software.

\*

\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,

\* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF  
\* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.  
\* IN NO EVENT  
SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY  
\* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,  
\* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE  
\* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.  
\*\*\*\*\*  
]]

Found in path(s):

\* /lua-5-4-7-1-tar-gz/lua-5.4.7/testes/all.lua

No license file was found, but licenses were detected in source scan.

&copy; 2024 Lua.org, PUC-Rio. All rights reserved.

header = string.gsub(header, "All rights reserved",

Found in path(s):

\* /lua-5-4-7-1-tar-gz/lua-5.4.7/manual/2html

## 1.70 curl 8.14.1-r2

### 1.70.1 Available under license :

Copyright (c) <year> <owner>.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## COPYRIGHT AND PERMISSION NOTICE

Copyright (C) Daniel Stenberg, <daniel@haxx.se>, and many contributors, see the THANKS file.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this

Software without prior written authorization of the copyright holder.

BSD-4-Clause (University of California-Specific)

Copyright [various years] The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2025, Daniel Stenberg, <daniel@haxx.se>, and many contributors, see the THANKS file.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

## 1.71 cpp 14.2.0-r6

## 1.71.1 Available under license :

### GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we

want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and

distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are



prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6.

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING

OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),  
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands

`show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

-----  
This program, "bzip2", the associated library "libbzip2", and all documentation, are copyright (C) 1996-2010 Julian R Seward. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, jseward@bzip.org  
bzip2/libbzip2 version 1.0.6 of 6 September 2010

-----

GNU LESSER GENERAL PUBLIC LICENSE  
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts  
as the successor of the GNU Library Public License, version 2, hence  
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your  
freedom to share and change it. By contrast, the GNU General Public  
Licenses are intended to guarantee your freedom to share and change  
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some  
specially designated software packages--typically libraries--of the  
Free Software Foundation and other authors who decide  
to use it. You  
can use it too, but we suggest you first think carefully about whether  
this license or the ordinary General Public License is the better  
strategy to use in any particular case, based on the explanations  
below.

When we speak of free software, we are referring to freedom of use,  
not price. Our General Public Licenses are designed to make sure that

you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a

combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

## GNU LESSER GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or

other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:



- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary

## GNU General Public

License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The

threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object

code for the work under the terms of Section 6.

Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly

with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally

distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any

such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO

WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,

but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!

The libffi source distribution contains certain code that is not part of libffi, and is only used as tooling to assist with the building and testing of libffi. This includes the msvcc.sh script used to wrap the Microsoft compiler with GNU compatible command-line options, make\_sunver.pl, and the libffi test code distributed in the testsuite/libffi.bhaible directory. This code is distributed with libffi for the purpose of convenience only, and libffi is in no way derived from this code.

msvcc.sh and testsuite/libffi.bhaible are both distributed under the terms of the GNU GPL version 2, as below.

## GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### Preamble

The licenses for most software



are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and

modification follow.

GNU GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

You

are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will

be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest

possible use to the public, the best way to achieve  
this is to make it  
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest  
to attach them to the start of each source file to most effectively  
convey the exclusion of warranty; and each file should have at least  
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify  
it under the terms of the GNU General Public License as published by  
the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more  
details.

You should have received a copy of the GNU General Public License along  
with this program; if not, write to the Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this  
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate  
parts of the General Public License. Of course, the commands you use may  
be called something other than `show w' and `show c'; they could even be  
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work  
as a programmer) or your  
school, if any, to sign a "copyright disclaimer" for the program, if  
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program

`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.



"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such

obligations, You may act only  
on Your own behalf and on Your sole responsibility, not on behalf  
of any other Contributor, and only if You agree to indemnify,  
defend, and hold each Contributor harmless for any liability  
incurred by, or claims asserted against, such Contributor by reason  
of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following  
boilerplate notice, with the fields enclosed by brackets "[ ]"  
replaced with your own identifying information. (Don't include  
the brackets!) The text should be enclosed in the appropriate  
comment syntax for the file format. We also recommend that a  
file or class name and description of purpose be included on the  
same "printed page" as the copyright notice for easier  
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

### ---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions  
of this Software are embedded into an Object form of such source code, you  
may redistribute such embedded portions in such Object form without complying  
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled  
forms of this Software with  
software that is licensed under the GPLv2 ("Combined Software") and if a  
court of competent jurisdiction determines that the patent provision (Section  
3), the indemnity provision (Section 9) or other Section of the License  
conflicts with the conditions of the GPLv2, you may retroactively and

prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

=====

Software from third parties included in the LLVM Project:

=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or `LICENSE` file at the top containing the specific license and restrictions which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every file.

=====

Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):

=====

The software contained in this directory tree is dual licensed under both the University of Illinois "BSD-Like" license and the MIT license. As a user of this code you may choose to use it under either license. As a contributor, you agree to allow your code to be used under both.

Full text of the relevant licenses is included below.

=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2017-2019 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:  
Threading Runtimes Team  
Intel Corporation  
<http://www.intel.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- \* Neither the names of Intel Corporation Threading Runtimes Team nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====

Copyright (c) 2017-2019 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

GNU LESSER GENERAL PUBLIC LICENSE  
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.



In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

## GNU LESSER GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of

running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves,

then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy

from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License.

Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6.

Any

executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work

during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed

under the terms of the  
Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying  
or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with

this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute  
so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many  
people have made  
generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public  
License from time to time.  
Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

##### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest



possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!  
Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the

Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS

FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,  
each Contributor hereby grants to You a perpetual,  
worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer,

and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2011 University of Illinois at Urbana-Champaign. All rights reserved.

Developed by: Polaris Research Group  
University of Illinois at Urbana-Champaign  
<http://polaris.cs.uiuc.edu>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

1. Redistributions of source code must retain the above copyright notice,

this list of conditions and the following disclaimers.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
3. Neither the names of Polaris Research Group, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

#### GCC RUNTIME LIBRARY EXCEPTION

Version 3.1, 31 March 2009

Copyright (C) 2009 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This GCC Runtime Library Exception ("Exception") is an additional permission under section 7 of the GNU General Public License, version 3 ("GPLv3"). It applies to a given file (the "Runtime Library") that bears a notice placed by the copyright holder of the file stating that the file is governed by GPLv3 along with this Exception.

When you use GCC to compile a program, GCC may combine portions of certain GCC header files and runtime libraries with the compiled program. The purpose of this Exception is to allow compilation of non-GPL (including proprietary) programs to use, in this way, the header files and runtime libraries covered by this Exception.

#### 0. Definitions.

A file is an "Independent Module" if it either requires the Runtime Library for execution after a Compilation Process, or makes use of an interface provided by the Runtime Library, but is not otherwise based on the Runtime Library.

"GCC" means a version of the GNU Compiler Collection, with or without modifications, governed by version 3 (or a specified later version) of the GNU General Public License (GPL) with the option of using any

subsequent versions published by the FSF.

"GPL-compatible Software" is software whose conditions of propagation, modification and use would permit combination with GCC in accord with the license of GCC.

"Target Code" refers to output from any compiler for a real or virtual target processor architecture, in executable form or suitable for input to an assembler, loader, linker and/or execution phase. Notwithstanding that, Target Code does not include data in any format that is used as a compiler intermediate representation, or used for producing a compiler intermediate representation.

The "Compilation Process" transforms code entirely represented in non-intermediate languages designed for human-written code, and/or in Java Virtual Machine byte code, into Target Code. Thus, for example, use of source code generators and preprocessors need not be considered part of the Compilation Process, since the Compilation Process can be understood as starting with the output of the generators or preprocessors.

A Compilation Process is "Eligible" if it is done using GCC, alone or with other GPL-compatible software, or if it is done without using any work based on GCC. For example, using non-GPL-compatible Software to optimize any GCC intermediate representations would not qualify as an Eligible Compilation Process.

#### 1. Grant of Additional Permission.

You have permission to propagate a work of Target Code formed by combining the Runtime Library with Independent Modules, even if such propagation would otherwise violate the terms of GPLv3, provided that all Target Code was generated by Eligible Compilation Processes. You may then convey such a combination under terms of your choice, consistent with the licensing of the Independent Modules.

#### 2. No Weakening of GCC Copyleft.

The availability of this Exception does not imply any general presumption that third-party software is unaffected by the copyleft requirements of the license of GCC.

libffi - Copyright (c) 1996-2021 Anthony Green, Red Hat, Inc and others.  
See source files for details.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the



``Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED ``AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

GNU Free Documentation License  
Version 1.2, November 2002

Copyright (C) 2000-2024 Free Software Foundation, Inc.  
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or

whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the

general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for

automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties; any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3.

### COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an

Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

#### 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for

public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights

of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.



## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

### ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.  
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

```
/* Definitions for BSD assembler syntax for Intel 386
   (actually AT&T syntax for insns and operands,
   adapted to BSD conventions for symbol names and debugging.)
   Copyright (C) 1988-2024 Free Software Foundation, Inc.
```

This file is part of GCC.

GCC is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3, or (at your option) any later version.

GCC is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with GCC; see the file COPYING3. If not see <<http://www.gnu.org/licenses/>>. \*/

```
/* Use the Sequent Symmetry assembler syntax. */
```

```
/* Define the syntax of pseudo-ops, labels and comments. */
```

```
/* Prefix for internally generated assembler
   labels. If we aren't using
   underscores, we are using prefix `.'s to identify labels that should
   be ignored, as in `i386/gas.h' --karl@cs.umb.edu */
```

```
#define LPREFIX "L"
```

```
/* Assembler pseudos to introduce constants of various size. */
```

```
#define ASM_BYTE "\t.byte\t"
#define ASM_SHORT "\t.word\t"
#define ASM_LONG "\t.long\t"
#define ASM_QUAD "\t.quad\t" /* Should not be used for 32bit compilation. */
```

```
/* This was suggested, but it shouldn't be right for debugger output. -- RMS
   #define ASM_OUTPUT_SOURCE_FILENAME(FILE, NAME) */
```

```
/* Define the syntax of labels and symbol definitions/declarations. */
```

```
/* This is how to output an assembler line
   that says to advance the location counter by SIZE bytes. */
```

```

#define ASM_OUTPUT_SKIP(FILE,SIZE) \
    fprintf (FILE, "\t.space " HOST_WIDE_INT_PRINT_UNSIGNED"\n", (SIZE))

/* Define the syntax of labels and symbol definitions/declarations. */

/* This says how to output an assembler line
   to define a global common symbol. */

#define
    ASM_OUTPUT_COMMON(FILE, NAME, SIZE, ROUNDED) \
    ( fputs (".comm ", (FILE)), \
      assemble_name ((FILE), (NAME)), \
      fprintf ((FILE), ",%u\n", (int)(ROUNDED)))

/* This says how to output an assembler line
   to define a local common symbol. */

#define ASM_OUTPUT_LOCAL(FILE, NAME, SIZE, ROUNDED) \
    ( fputs (".lcomm ", (FILE)), \
      assemble_name ((FILE), (NAME)), \
      fprintf ((FILE), ",%u\n", (int)(ROUNDED)))

#ifdef HAVE_GAS_LCOMM_WITH_ALIGNMENT
#define ASM_OUTPUT_ALIGNED_LOCAL(FILE, NAME, SIZE, ALIGNMENT) \
    ( fputs (".lcomm ", (FILE)), \
      assemble_name ((FILE), (NAME)), \
      fprintf ((FILE), ",%u,%u\n", (int)(SIZE), (int)(ALIGNMENT) / BITS_PER_UNIT))
#endif

/* This is how to output an assembler line
   that says to advance the location counter
   to a multiple of 2**LOG bytes. */

#define ASM_OUTPUT_ALIGN(FILE,LOG) \
    if ((LOG)!=0) fprintf ((FILE), "\t.align %d\n", (LOG))

/* This is how to store into the string BUF
   the symbol_ref name of an internal numbered
   label where
   PREFIX is the class of label and NUM is the number within the class.
   This is suitable for output with `assemble_name'. */

#define ASM_GENERATE_INTERNAL_LABEL(BUF,PREFIX,NUMBER) \
    sprintf ((BUF), "%s%d", (PREFIX), (long)(NUMBER))

/* The prefix to add to user-visible assembler symbols. */

#define USER_LABEL_PREFIX "_"

```

/\* Copyright (C) 2013-2024 Free Software Foundation, Inc.

This file is part of GCC.

modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2, or (at your option) any later version.

In addition to the permissions in the GNU Library General Public License, the Free Software Foundation gives you unlimited permission to link the compiled version of this file into combinations with other programs, and to distribute those combinations without any restriction coming from the use of this file. (The Library Public License restrictions do apply in other respects; for example, they cover modification of the file, and distribution when not linked into a combined executable.)

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street - Fifth Floor, Boston, MA 02110-1301, USA. \*/

```
#ifndef __VTV_H__
#define __VTV_H__
```

```
/* We could have used an enumeration here but it just makes it more
   difficult for the compiler to generate a call to this. These are
   used as arguments to the function __VLTChangePermission, declared
   below. */
```

```
#define __VLTP_READ_ONLY 0
#define __VLTP_READ_WRITE 1
```

```
#ifdef __cplusplus
extern "C" void __VLTChangePermission (int);
#else
extern void __VLTChangePermission (int);
#endif
```

```
#ifdef BIG_PAGE_SIZE
/* TODO - Replace '4096' below with correct big page size. */
#define VTV_PAGE_SIZE 4096
```

```

#else
#if defined(__sun__) && defined(__svr4__) && defined(__sparc__)
#define VTV_PAGE_SIZE 8192
#elif defined(__loongarch_lp64)
/* The page size is configurable by the kernel to
be 4, 16 or 64 KiB.
For now, only the default page size of 16KiB is supported. */
#define VTV_PAGE_SIZE 16384
#else
#define VTV_PAGE_SIZE 4096
#endif
#endif

```

```

#endif /* __VTV_H__ */
# This source code was written by the Go contributors.
# The master list of contributors is in the main Go distribution,
# visible at http://tip.golang.org/CONTRIBUTORS.
Copyright (c) 2009 The Go Authors. All rights reserved.

```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### Preamble

The GNU General Public License is a free, copyleft license for  
software and other kinds of works.

The licenses for most software and other practical works are designed  
to take away your freedom to share and change the works. By contrast,  
the GNU General Public License is intended to guarantee your freedom to  
share and change all versions of a program--to make sure it remains free  
software for all its users. We, the Free Software Foundation, use the  
GNU General Public License for most of our software; it applies also to  
any other work released this way by its authors. You can apply it to  
your programs, too.

When we speak of free software, we are referring to  
freedom, not  
price. Our General Public Licenses are designed to make sure that you  
have the freedom to distribute copies of free software (and charge for  
them if you wish), that you receive source code or can get it if you  
want it, that you can change the software or use pieces of it in new  
free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you  
these rights or asking you to surrender the rights. Therefore, you have  
certain responsibilities if you distribute copies of the software, or if  
you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether  
gratis or for a fee, you must pass on to the recipients the same  
freedoms that you received. You must make sure that they, too, receive  
or can get the source code. And you must show them these terms so they  
know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1)

assert copyright on the software, and (2) offer you this License  
giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains  
that there is no warranty for this free software. For both users' and  
authors' sake, the GPL requires that modified versions be marked as

changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS

### 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without

permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

#### 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The

"System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally



available free

programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

## 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered

by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed;

section 10

makes it unnecessary.

## 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or

similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

#### 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

#### 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

## 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated

place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the

User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

## 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or

limiting liability differently from the terms of sections 15 and 16 of this License; or

- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on

those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

## 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or

modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

#### 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

#### 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that

transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights

granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

#### 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this



License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

## 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may

not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

### 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

### 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

### 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS"

WITHOUT WARRANTY

OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17.

Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by

the Free Software Foundation, either version 3 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.

You should have received a copy of the GNU General Public License  
along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short  
notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>  
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate  
parts of the General Public License. Of course, your program's commands  
might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school,  
if any, to sign a "copyright disclaimer" for the program, if necessary.  
For more information on this, and how to apply and follow the GNU GPL, see  
<<http://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program  
into proprietary programs. If your program is a subroutine library, you  
may consider it more useful to permit linking proprietary applications with  
the library. If this is what you want to  
do, use the GNU Lesser General  
Public License instead of this License. But first, please read  
<<http://www.gnu.org/philosophy/why-not-lgpl.html>>.  
The files gnupng and gnu.eps are part of GNU Modula-2.

Copyright (C) 2005-2024 Free Software Foundation, Inc.

This file is part of GNU Modula-2.

GNU Modula-2 is free software; you can redistribute it and/or modify  
it under the terms of the GNU General Public License as published by  
the Free Software Foundation; either version 3, or (at your option)  
any later version.

GNU Modula-2 is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with GNU Modula-2; see the file COPYING. If not, write to the Free Software Foundation, 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA. \*/

## GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>  
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

### 0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the

object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

#### 1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

#### 2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

#### 3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

#### 4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse

engineering for debugging such modifications, if you also do each of the following:

- a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the Combined Work with a copy of the GNU GPL and this license document.
- c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.
- d) Do one of the following:
  - 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.
  - 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.
- e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

## 5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
- b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

#### 6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

#### GCC RUNTIME LIBRARY EXCEPTION

Version 3.1, 31 March 2009

Copyright (C) 2009-2024 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This GCC Runtime Library Exception ("Exception") is an additional



permission under section 7 of the GNU General Public License, version 3 ("GPLv3"). It applies to a given file (the "Runtime Library") that bears a notice placed by the copyright holder of the file stating that the file is governed by GPLv3 along with this Exception.

When you use GCC to compile a program, GCC may combine portions of certain GCC header files and runtime libraries with the compiled program. The purpose of this Exception is to allow compilation of non-GPL (including proprietary) programs to use, in this way, the header files and runtime libraries covered by this Exception.

## 0. Definitions.

A file is an "Independent Module" if it either requires the Runtime Library for execution after a Compilation Process, or makes use of an interface provided by the Runtime Library, but is not otherwise based on the Runtime Library.

"GCC" means a version of the GNU Compiler Collection, with or without modifications, governed by version 3 (or a specified later version) of the GNU General Public License (GPL) with the option of using any subsequent versions published by the FSF.

"GPL-compatible Software" is software whose conditions of propagation, modification and use would permit combination with GCC in accord with the license of GCC.

"Target Code" refers to output from any compiler for a real or virtual target processor architecture, in executable form or suitable for input to an assembler, loader, linker and/or execution phase. Notwithstanding that, Target Code does not include data in any format that is used as a compiler intermediate representation, or used for producing a compiler intermediate representation.

The "Compilation Process" transforms code entirely represented in non-intermediate languages designed for human-written code, and/or in Java Virtual Machine byte code, into Target Code. Thus, for example, use of source code generators and preprocessors need not be considered part of the Compilation Process, since the Compilation Process can be understood as starting with the output of the generators or preprocessors.

A Compilation Process is "Eligible" if it is done using GCC, alone or with other GPL-compatible software, or if it is done without using any work based on GCC. For example, using non-GPL-compatible Software to optimize any GCC intermediate representations would not qualify as an

Eligible Compilation Process.

#### 1. Grant of Additional Permission.

You have permission to propagate a work of Target Code formed by combining the Runtime Library with Independent Modules, even if such propagation would otherwise violate the terms of GPLv3, provided that all Target Code was generated by Eligible Compilation Processes.

You may then convey such a combination under terms of your choice, consistent with the licensing of the Independent Modules.

#### 2. No Weakening of GCC Copyleft.

The availability of this Exception does not imply any general presumption that third-party software is unaffected by the copyleft requirements of the license of GCC.

=====  
compiler\_rt License

=====  
The compiler\_rt library is dual licensed under both the University of Illinois "BSD-Like" license and the MIT license. As a user of this code you may choose to use it under either license. As a contributor, you agree to allow your code to be used under both.

Full text of the relevant licenses is included below.

=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2009-2012 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

<http://llvm.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction,

including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- \* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====

Copyright (c) 2009-2012 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====  
Copyrights and Licenses for Third Party Software Distributed with LLVM:  
=====

The LLVM software contains code written by third parties. Such software will have its own individual LICENSE.TXT file in the directory in which it appears. This file will describe the copyrights, license, and restrictions which apply to that code.

The disclaimer of warranty in the University of Illinois Open Source License applies to all code in the LLVM Distribution, and nothing in any of the other licenses gives permission to use the names of the LLVM Team or the University of Illinois to endorse or promote products derived from this Software.

The following pieces of software have additional or alternate copyrights, licenses, and/or restrictions:

| Program       | Directory                      |
|---------------|--------------------------------|
| -----         | -----                          |
| mach_override | lib/interception/mach_override |

UNICODE, INC. LICENSE AGREEMENT - DATA FILES AND SOFTWARE

Unicode Data Files include all data files under the directories <http://www.unicode.org/Public/>, <http://www.unicode.org/reports/>, and <http://www.unicode.org/cldr/data/>. Unicode Data Files do not include PDF online code charts under the directory <http://www.unicode.org/Public/>. Software includes any source code published in the Unicode Standard or under the directories <http://www.unicode.org/Public/>, <http://www.unicode.org/reports/>, and <http://www.unicode.org/cldr/data/>.

NOTICE TO USER: Carefully read the following legal agreement. BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.'S DATA FILES ("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"), YOU UNEQUIVOCALLY ACCEPT, AND AGREE TO BE BOUND BY, ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE THE DATA FILES OR SOFTWARE.

#### COPYRIGHT AND PERMISSION NOTICE

Copyright 1991-2013 Unicode, Inc. All rights reserved. Distributed under the Terms of Use in <http://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software")

to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that (a) the above copyright notice(s) and this permission notice appear with all copies of the Data Files or Software, (b) both the above copyright notice(s) and this permission notice appear in associated documentation, and (c) there is clear notice in each modified Data File or in the Software as well as in the documentation associated with the Data File(s) or Software that the data or software has been modified.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

Copyright (c) 2019 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.72 libnetfilter-cttimeout 1.0.1-r2

### 1.72.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

675 Mass Ave, Cambridge, MA 02139, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the

source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you

conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or



collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6.

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the

integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES

PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

#### Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

## 1.73 bzip2 1.0.8-r6

### 1.73.1 Available under license :

-----

This program, "bzip2", the associated library "libbzip2", and all documentation, are copyright (C) 1996-2019 Julian R Seward. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, [jseward@acm.org](mailto:jseward@acm.org)  
bzip2/libbzip2 version 1.0.8 of 13 July 2019

## 1.74 pyproject-hooks 1.2.0

### 1.74.1 Available under license :

The MIT License (MIT)

Copyright (c) 2017 Thomas Kluyver

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.  
MIT

## 1.75 libnetfilter-cthelper 1.0.1-r2

### 1.75.1 Available under license :

Format: <http://dep.debian.net/deps/dep5>  
Upstream-Name: libnetfilter-cthelper  
Source: [http://www.netfilter.org/projects/libnetfilter\\_cthelper/index.html](http://www.netfilter.org/projects/libnetfilter_cthelper/index.html)

Files: \*  
Copyright: 2012 Pablo Neira Ayuso <pablo@netfilter.org>  
License: GPL-2.0+

Files: debian/\*  
Copyright: 2013 Alexander Wirt <formorer@debian.org>  
License: GPL-2.0+

License: GPL-2.0+  
This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 2 of the License, or (at your option) any later version.

.  
This package is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

.  
You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

.  
On Debian systems, the complete text of the GNU General Public License version 2 can be found in "/usr/share/common-licenses/GPL-2".

GNU GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
675 Mass Ave, Cambridge, MA 02139, USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.



Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections

1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and

all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6.

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the

original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER

PROGRAMS),  
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE  
POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may

be called something other than `show w` and `show c`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

## 1.76 python-pip 25.1.1

### 1.76.1 Available under license :

#### A. HISTORY OF THE SOFTWARE

=====

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see <http://www.cwi.nl>) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see <http://www.cnri.reston.va.us>) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations (now Zope Corporation, see <http://www.zope.com>). In 2001, the Python Software Foundation (PSF, see <http://www.python.org/psf/>) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation is a sponsoring member of  
of  
the PSF.

All Python releases are Open Source (see <http://www.opensource.org> for the Open Source Definition). Historically, most, but not all, Python

releases have also been GPL-compatible; the table below summarizes the various releases.

| Release        | Derived<br>from | Year      | Owner      | GPL-<br>compatible? (1) |
|----------------|-----------------|-----------|------------|-------------------------|
| 0.9.0 thru 1.2 |                 | 1991-1995 | CWI        | yes                     |
| 1.3 thru 1.5.2 | 1.2             | 1995-1999 | CNRI       | yes                     |
| 1.6            | 1.5.2           | 2000      | CNRI       | no                      |
| 2.0            | 1.6             | 2000      | BeOpen.com | no                      |
| 1.6.1          | 1.6             | 2001      | CNRI       | yes (2)                 |
| 2.1            | 2.0+1.6.1       | 2001      | PSF        | no                      |
| 2.0.1          | 2.0+1.6.1       | 2001      | PSF        | yes                     |
| 2.1.1          | 2.1+2.0.1       | 2001      | PSF        | yes                     |
| 2.2            | 2.1.1           | 2001      | PSF        | yes                     |
| 2.1.2          | 2.1.1           | 2002      | PSF        | yes                     |
| 2.1.3          |                 |           |            |                         |
| 2.1.2          | 2002            | PSF       | yes        |                         |
| 2.2.1          | 2.2             | 2002      | PSF        | yes                     |
| 2.2.2          | 2.2.1           | 2002      | PSF        | yes                     |
| 2.2.3          | 2.2.2           | 2003      | PSF        | yes                     |
| 2.3            | 2.2.2           | 2002-2003 | PSF        | yes                     |
| 2.3.1          | 2.3             | 2002-2003 | PSF        | yes                     |
| 2.3.2          | 2.3.1           | 2002-2003 | PSF        | yes                     |
| 2.3.3          | 2.3.2           | 2002-2003 | PSF        | yes                     |
| 2.3.4          | 2.3.3           | 2004      | PSF        | yes                     |
| 2.3.5          | 2.3.4           | 2005      | PSF        | yes                     |
| 2.4            | 2.3             | 2004      | PSF        | yes                     |
| 2.4.1          | 2.4             | 2005      | PSF        | yes                     |
| 2.4.2          | 2.4.1           | 2005      | PSF        | yes                     |
| 2.4.3          | 2.4.2           | 2006      | PSF        | yes                     |
| 2.4.4          | 2.4.3           | 2006      | PSF        | yes                     |
| 2.5            | 2.4             | 2006      | PSF        | yes                     |
| 2.5.1          | 2.5             | 2007      | PSF        | yes                     |
| 2.5.2          | 2.5.1           | 2008      | PSF        | yes                     |
| 2.5.3          | 2.5.2           | 2008      | PSF        | yes                     |
| 2.6            | 2.5             | 2008      | PSF        | yes                     |
| 2.6.1          | 2.6             | 2008      | PSF        | yes                     |
| 2.6.2          | 2.6.1           | 2009      | PSF        | yes                     |
| 2.6.3          | 2.6.2           | 2009      | PSF        | yes                     |
| 2.6.4          | 2.6.3           | 2009      | PSF        | yes                     |
| 2.6.5          | 2.6.4           | 2010      | PSF        | yes                     |
| 3.0            | 2.6             | 2008      | PSF        | yes                     |
| 3.0.1          | 3.0             | 2009      | PSF        | yes                     |
| 3.1            | 3.0.1           | 2009      | PSF        | yes                     |
| 3.1.1          | 3.1             | 2009      | PSF        | yes                     |
| 3.1.2          | 3.1             | 2010      | PSF        | yes                     |
| 3.2            | 3.1             | 2010      | PSF        | yes                     |



Footnotes:

- (1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.
- (2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

## B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

=====

### PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

-----

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.
2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010 Python Software Foundation; All Rights Reserved" are retained in Python alone or in any derivative version prepared by Licensee.
3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.
4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS

OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

#### BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

#### BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3.  
BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR

IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at <http://www.pythonlabs.com/logos.html> may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

#### CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

-----

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative

version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the Internet using the following  
URL: <http://hdl.handle.net/1895.22/1013>".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6.  
This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or

services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

## CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

-----

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The MIT License (MIT)

Copyright (c) 2017 Thomas Kluyver

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR

IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

SPDX-License-Identifier: MIT

SPDX-FileCopyrightText: 2021 Taneli Hukkinen

Licensed to PSF under a Contributor Agreement.

Copyright (c) 2020 Will McGugan

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

#### A. HISTORY OF THE SOFTWARE

=====

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see <https://www.cwi.nl>) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see <https://www.cnri.reston.va.us>) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations, which became Zope Corporation. In 2001, the Python Software Foundation (PSF, see <https://www.python.org/psf/>) was formed, a non-profit organization created specifically to own Python-related Intellectual Property.

Zope Corporation was a sponsoring member of the PSF.

All

Python releases are Open Source (see <https://opensource.org> for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

| Release        | Derived from | Year      | Owner      | GPL-compatible? (1) |
|----------------|--------------|-----------|------------|---------------------|
| 0.9.0 thru 1.2 |              | 1991-1995 | CWI        | yes                 |
| 1.3 thru 1.5.2 | 1.2          | 1995-1999 | CNRI       | yes                 |
| 1.6            | 1.5.2        | 2000      | CNRI       | no                  |
| 2.0            | 1.6          | 2000      | BeOpen.com | no                  |
| 1.6.1          | 1.6          | 2001      | CNRI       | yes (2)             |
| 2.1            | 2.0+1.6.1    | 2001      | PSF        | no                  |
| 2.0.1          | 2.0+1.6.1    | 2001      | PSF        | yes                 |
| 2.1.1          | 2.1+2.0.1    | 2001      | PSF        | yes                 |
| 2.1.2          | 2.1.1        | 2002      | PSF        | yes                 |
| 2.1.3          | 2.1.2        | 2002      | PSF        | yes                 |
| 2.2 and above  | 2.1.1        | 2001-now  |            |                     |
| PSF            |              |           |            | yes                 |

Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

## B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

=====

Python software and documentation are licensed under the Python Software Foundation License Version 2.

Starting with Python 3.8.6,

examples, recipes, and other code in the documentation are dual licensed under the PSF License Version 2 and the Zero-Clause BSD license.

Some software incorporated into Python is under different licenses. The licenses are listed with code falling under that license.

## PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

-----

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.
2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023 Python Software Foundation; All Rights Reserved" are retained in Python alone or in any derivative version prepared by Licensee.
3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.
4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.
5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.
6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.



7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

#### BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

-----

#### BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at <http://www.pythonlabs.com/logos.html> may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

#### CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

-----

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the internet using the following URL: <http://hdl.handle.net/1895.22/1013>".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of

the changes made  
to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

-----

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER

RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

#### ZERO-CLAUSE BSD LICENSE FOR CODE IN THE PYTHON DOCUMENTATION

-----  
Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

This package contains a modified version of ca-bundle.crt:

ca-bundle.crt -- Bundle of CA Root Certificates

This is a bundle of X.509 certificates of public Certificate Authorities (CA). These were automatically extracted from Mozilla's root certificates file (certdata.txt). This file can be found in the mozilla source tree: <https://hg.mozilla.org/mozilla-central/file/tip/security/nss/lib/ckfw/builtins/certdata.txt> It contains the certificates in PEM format and therefore can be directly used with curl / libcurl / php\_curl, or with an Apache+mod\_ssl webserver for SSL client authentication. Just configure this file as the SSLCertificateFile.#

\*\*\*\*\* BEGIN LICENSE BLOCK \*\*\*\*\*

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain

one at <http://mozilla.org/MPL/2.0/>.

\*\*\*\*\* END LICENSE BLOCK \*\*\*\*\*

@(#) \$RCSfile: certdata.txt,v \$ \$Revision: 1.80 \$ \$Date: 2011/11/03 15:11:58 \$  
MIT License

Copyright (c) 2024-present Stephen Rosen <[sirosen0@gmail.com](mailto:sirosen0@gmail.com)>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity

exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,



work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

#### END OF TERMS AND CONDITIONS

This software is made available under the terms of *\*either\** of the licenses found in LICENSE.APACHE or LICENSE.BSD. Contributions to this software is made under the terms of *\*both\** these licenses.

MIT License

Copyright (c) 2008-2020 Andrey Petrov and contributors (see CONTRIBUTORS.txt)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN

THE SOFTWARE.

MIT License

Copyright (c) 2021 Taneli Hukkinen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2010-202x The platformdirs developers

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT

OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

BSD 3-Clause License

Copyright (c) 2013-2024, Kim Davies and contributors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2018, Tzu-ping Chung <uranusjr@gmail.com>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR

ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) Donald Stuftt and individual contributors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT

The MIT License (MIT)

Copyright (c) 2022 Seth Michael Larson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT

OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2012-2021 Eric Larson

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2008-present The pip developers (see AUTHORS.txt file)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2006-2022 by the respective authors (see AUTHORS file).

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- \* Redistributions in binary form must reproduce the above copyright

notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.  
Copyright (C) 2008-2011 INADA Naoki <songofacandy@gmail.com>

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one



of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You"  
(or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants

to You a perpetual,  
worldwide, non-exclusive, no-charge, royalty-free, irrevocable  
(except as stated in this section) patent license to make, have made,  
use, offer to sell, sell, import, and otherwise transfer the Work,  
where such license applies only to those patent claims licensable  
by such Contributor that are necessarily infringed by their  
Contribution(s) alone or by combination of their Contribution(s)  
with the Work to which such Contribution(s) was submitted. If You  
institute patent litigation against any entity (including a  
cross-claim or counterclaim in a lawsuit) alleging that the Work  
or a Contribution incorporated within the Work constitutes direct  
or contributory patent infringement, then any patent licenses  
granted to You under this License for that Work shall terminate  
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the  
Work

or Derivative Works thereof in any medium, with or without  
modifications, and in Source or Object form, provided that You  
meet the following conditions:

- (a) You must give any other recipients of the Work or  
Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices  
stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works  
that You distribute, all copyright, patent, trademark, and  
attribution notices from the Source form of the Work,  
excluding those notices that do not pertain to any part of  
the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its  
distribution, then any Derivative Works that You distribute must  
include a readable copy of the attribution notices contained  
within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one  
of the following places: within a NOTICE text file distributed  
as part of the Derivative Works; within the Source form or  
documentation, if provided along with the Derivative Works; or,  
within a display generated by the Derivative Works, if and  
wherever such third-party notices normally appear. The contents  
of the NOTICE file are for informational purposes only and  
do not modify the License. You may add Your own attribution  
notices within Derivative Works that You distribute, alongside  
or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## 1.77 zsh 5.9-r5

### 1.77.1 Available under license :

```
#autoload
```

```
# Options:
```

```
#
```

```
# -0    include jid 0 as a match for the host system
```

```

# -o param jail parameter to complete instead of jid -
#           e.g. name, path, ip4.addr, host.hostname

local addhost host param desc=1
local -a jails args expl
zparseopts -D -K -E 0=addhost o:=param
param=${param[2]:-name}

jails=( ${$(f)"$_call_program jails jls $param name)"/ /:} )
case $param in
jid) host=0 ;;
name)
    host=0
    desc=0
;;
path)
    host=/
    args=( -M 'r:/=* r:|=*' )
;;
ip4.addr) args=( -M 'r:|=* r:|=*' ) ;;
esac
[[ -n $addhost && -n $host ]] && jails+=( "$host:$HOST" )

if (( desc )); then
    _describe -t jails jail jails "$@" "$args[@]"
else
    _wanted jails expl jail compadd "$@" "$args[@]" - ${jails%:*}
fi
#autoload

local points=(
'c[trace system calls]'
'i[trace I/O]'
'n[trace namei translations]'
's[trace signal processing]'
'u[trace user data]'
'+[trace the default points]'
)

case $OSTYPE in
dragonfly*|freebsd*|netbsd*)
    points+=(
        'w[context switches]'
    )
    ;|
freebsd*|openbsd*)
    points+=(
        't[trace various structures]'
    )

```

```

)
;|
freebsd*)
points+=(
    'f[trace page faults]'
    'p[trace capability check failures]'
    'y[trace sysctl(3) requests]'
)
;;
netbsd*)
points+=(
    'A[trace all tracepoints]'
    'a[trace exec arguments]'
    'e[trace emulation changes]'
    'f[trace open file descriptors after exec]'
    'S[trace MIB access (sysctl)]'
    'v[trace exec environment]'
    '-[do not trace following trace points]'
)
;;
openbsd*)
points+=(
    'p[trace violation of pledge(2) restrictions]'
    'x[trace argument vector in execve(2)]'
    'X[trace environment in execve(2)]'
)
;;
esac

_values -s " 'ktrace point' $points
#autoload
#
# device types on FreeBSD/DragonFly
# (for commands using devstat_buildmatch(), such as iostat and vmstat)
#
local -a d i types

d=( da sa printer proc worm cd scanner optical changer
    comm array enclosure floppy)
i=( IDE SCSI other )
types=(
    "($d)da[direct access devices]"
    "($d)sa[sequential access devices]"
    "($d)printer[printers]"
    "($d)proc[processor devices]"
    "($d)worm[write once read multiple devices]"
    "($d)cd[CD devices]"
    "($d)scanner[scanner devices]"

```



```

"($d)optical[optical memory devices]"
"($d)changer[medium changer devices]"
"($d)comm[communication devices]"
"($d)array[storage array devices]"
"($d)enclosure[enclosure services devices]"
"($d)floppy[floppy devices]"
"($i)IDE[Integrated Drive Electronics devices]"
"($i)SCSI[Small Computer System Interface devices]"
"($i)other[any other device interface]"
'pass[passthrough devices]'
)

_values -s , 'device type' $types
#autoload

local expl

_description architectures expl 'architecture'
compadd "$@" "$expl[@]" amd64 arm arm64 i386 mips powerpc riscv sparc64
#autoload

local expl

_description routing-domains expl 'routing domain'
compadd "$@" "$expl[@]" - ${$(M)}${(f)}${(_call_program routing-domains netstat -R)}:#Rdomain *}#Rdomain
}
#autoload

# Provides completion for file flags (formerly part of _chflags)

local curcontext=$curcontext su=$(( ! EUID || $_comp_priv_prefix ))
local -a context line state state_descr copts=( "${@}" ) flags flag_descs
local -A val_args

flag_descs+=(
  nodump nodump
  uappnd 'user append-only'
  uchg 'user immutable'
)

if (( su )); then
  flag_descs+=(
    arch archived
    sappnd 'system append-only'
    schg 'system immutable'
  )
fi

```

```

if [[ $OSTYPE = (darwin|dragonfly|freebsd|netbsd)* ]]; then
    flag_descs+=(opaque opaque)

if [[ $OSTYPE = darwin* ]]; then
    flag_descs+=(hidden hidden)
fi

if [[ $OSTYPE = (dragonfly|freebsd)* ]]; then
    flag_descs+=(uunlnk 'user undeletable')

    (( su )) &&
    flag_descs+=(sunlnk 'system undeletable')
fi

if [[ $OSTYPE = dragonfly* ]]; then
    flag_descs+=(
        cache cache
        nouhistory 'user nohistory'
    )

    (( su )) && flag_descs+=(
        noscache noscache
        noshistory 'system nohistory'
    )
fi

[[ $OSTYPE =
freebsd* ]] && flag_descs+=(
    uarch archive
    uhidden hidden
    uoffline offline
    urdonly 'DOS, Windows and CIFS readonly'
    ureparse 'Windows reparse point'
    usparse 'sparse file'
    usystem 'DOS, Windows and CIFS system'
)
fi

for 1 2 in $flag_descs; do
    if [[ $1 = no* ]]; then
        flags+=("${1#no}$1[set the $2 flag]"
            "$1){1#no}[unset the $2 flag]")
    else
        flags+=("(no$1)$1[set the $2 flag]"
            "$1)no$1[unset the $2 flag]")
    fi
done

```

```

_values -O copts -s , 'file flags' $flags
#autoload
# disk device names on BSDs
local -a disks

case $OSTYPE in
freebsd*)
    disks=( ${$(M){(f)"$(geom disk list)"}\:#Geom name\:.*}#\*: } )
;;
dragonfly*)
    disks=( $(sysctl -n kern.disks) )
;;
openbsd*)
    disks=( ${$(s,..)"$(sysctl -n hw.disknames)"}%\:.*} )
;;
netbsd*)
    disks=( $(sysctl -n hw.disknames) )
;;
esac

if (( $#disks )); then
    local expl
    _wanted disk-devices expl 'disk device' compadd "$@" $disks
    return
fi

return 1

```

Unless otherwise noted in the header of specific files, files in this distribution have the licence shown below.

However, note that certain shell functions are licensed under versions of the GNU General Public Licence. Anyone distributing the shell as a binary including those files needs to take account of this. Search shell functions for "Copyright" for specific copyright information. None of the core functions are affected by this, so those files may simply be omitted.

--

The Z Shell is copyright (c) 1992-2017 Paul Falstad, Richard Coleman, Zoltn Hidvgi, Andrew Main, Peter Stephenson, Sven Wischnowsky, and others. All rights reserved. Individual authors, whether or not specifically named, retain copyright in all changes; in what follows, they are referred to as 'the Zsh Development Group'. This is for convenience only and this body has no legal status. The Z shell is distributed under the following licence; any provisions made in individual files take precedence.

## Permission

is hereby granted, without written agreement and without licence or royalty fees, to use, copy, modify, and distribute this software and to distribute modified versions of this software for any purpose, provided that the above copyright notice and the following two paragraphs appear in all copies of this software.

In no event shall the Zsh Development Group be liable to any party for direct, indirect, special, incidental, or consequential damages arising out of the use of this software and its documentation, even if the Zsh Development Group have been advised of the possibility of such damage.

The Zsh Development Group specifically disclaim any warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The software provided hereunder is on an "as is" basis, and the Zsh Development Group have no obligation to provide maintenance, support, updates, enhancements, or modifications.

-----

## ZSH CONTRIBUTORS

-----

Zsh was originally written by Paul Falstad. Zsh is now maintained by the members of the zsh-workers mailing list <zsh-workers@zsh.org>. The development is currently coordinated by Peter Stephenson <pws@zsh.org>.

This file credits only the major contributors to the current release. See the ChangeLog files for a complete list of people who have submitted patches. Note that email addresses below and in the ChangeLog file are included for disambiguation purposes only, and are not guaranteed to be currently accurate. If you feel that you or someone else have been unfairly omitted from this list please mail the list <zsh-workers@zsh.org>.

## Version 4.3

-----

Contributors to the main shell include Clint Adams, Rocky Bernstein, Andrey Borzenkov, Dan Bullok, Marc Chantreux, Thorsten Dahlheimer, Wayne Davison, Lionel Flandrin, Michael Hwang, Oliver Kiddle, Greg Klanderman, Matthias Kopferman, Mikael Magnusson, Dan Nelson, Phil Pennock, Toby Peterson, R. Ramkumar, Franois Revol, Bart Schaefer, Tom Smetana, Joerg Sonnenberger, Peter Stephenson, Jon Strait, Takimoto Jun, River Tarnell, Frank Terbeck, Philippe Troin, Geoff Wing, Matt Wozniski.

The VCS\_Info function system was written by Frank Terbeck with contributions from Julien Langer.

Contributors to documentation and to completion and other shell functions include those above plus Nuno Alexandre, Ralph Amissah, Micah Anderson, arno, Frank Benkstein, Jay Berkenbilt, Frank Blendinger, Zack Cerza, Brendan Cully, Baptiste Daroussin, Danek Duvall, Ryan F, Sean Finney, Daniel Friesel, Alexey I. Froloff, Tobias Gruetzmacher, Pierre Habouzit, Richard Hartmann, Benjamin R. Haskell, Ag. D. Hatzimanikas, Pete Hollobon, Erik Johansson, Kohsuke Kawaguchi, Dough Kearns, Hannu Koivisto, Henryk Konsek, Chris Lamb, Vincent Lefevre, Roger Leigh, Eric Mangold, Max Mikhanosha, Arkadiusz Miskiewicz, Matthieu Moy, Scott Murray, Alexander Myodov, tpm Nmec, Kalle Olavi Niemitalo, Kazuhiro NISHIYAMA,

Omari Norman, Mustafa Oezkan, Tomasz Pala, Peter Palfrader, Carlos Phillips, Daniel Qarras, Jean-Baptiste Quenot, David Riebenbauer, Haakon Riiser, Ingo Rohlfs, Felix Rosencrantz, Simon Ruderich, Stephen Rger, William Scott, Kris Shannon, Jrg Sommer, Travis Spencer, Vincent Steman, Mrten Svantesson, Evgenii Terechkov, Magnus Therning, Bernhard Tittelbach, Ingmar Vanhassel, Markus Waldeck, Motoi Washida, Nikolai Weibull, Jesse Weinstein.

Version 4.2

-----

The contributor list is similar to that for 4.1.1.

Version 4.1.1

-----

Clint Adams <clint@zsh.org>: termcap/terminfo, completion, configuration, various modules, networking, prompts

Andrey Borzenkov <bor@zsh.org>: configuration, completion functions, cygwin support

Chmouel Boudjnah <chmouel@mandrakesoft.com>: Linux completion support

Wayne Davison <wayned@users.sourceforge.net>: shell history mechanism, arithmetic

Zvi Har'El <rl@math.technion.ac.il>: completion functions

Doug Kearns <djkea2@mugca.its.monash.edu.au>:  
completion functions

Oliver Kiddle <opk@zsh.org>: builtins, completion, parameters, prompts, standards support, tests

Andrew Main <zefram@zsh.org>: zle keymaps, games

Akinori Musha: BSD completion support

Felix Rosencrantz <f\_rosencrantz@yahoo.com>: completion, zle

Bart Schaefer <schaefer@zsh.org>: configuration, parameters, prompts, signals, syntax, tests, functions, sanity checking

Adam Spiers <adam@spiers.net>: completion and other functions

Peter Stephenson <pws@zsh.org>: configuration, zle, networking, syntax, completion

Tanaka Akira <akr@zsh.org>: completion functions

Philippe Troin <phil@fifi.org>: signals, process groups.

Geoff Wing <gcw@zsh.org>: archiving and distribution network, line editor display, configuration

Sven Wischnowsky <wischnow@zsh.org>: internal completion mechanism and low-level functions, line editor internals, syntax, wordcode mechanism,

Version 4.0.1

-----

Clint Adams <schizo@debian.org>: Completion functions, general configuration and module issues.

Andrej Borsenkow <bor@zsh.org>: Cygwin support, configuration, completion functions.

Chmouel Boudjnah <chmouel@mandrakesoft.com>: configuration support.

Wayne Davison <wayned@users.sourceforge.net>: History code.

Alexandre Duret-Lutz: completion functions and substitution code.

Oliver Kiddle <opk@zsh.org>: Modules and completion functions, AIX support.

Andrew Main <zefram@zsh.org>: Parameters and arithmetic, configuration.

Felix Rosencrantz <f\_rosencrantz@yahoo.com>: Test code and shell functions.

Bart Schaefer <schaefer@zsh.org>: bits of everything in all parts of code; general sanity control.

Adam Spiers <adam@spiers.net>: Completion and other functions, completion caching.

Peter Stephenson <pws@zsh.org>: configuration and installation, parameters, completion functions and configuration.

Tanaka Akira <akr@zsh.org>: Completion and test functions.

Geoff Wing <gcw@zsh.org>: FTP and mailing list archive

Sven Wischnowsky <wischnow@zsh.org>:

Most of the completion code, zpty,  
signal queueing, parameters, builtins, builtouts, built-here-and-theres;  
mend garden shed, rewire electricity, reinstall shutters, fix TV aerial in  
howling gale...

Other patches from E. Jay Birkenbilt, Jan Fedak, Ulrik Haugen, Paul  
Johnson, Fr. Br. George (George V Kouryachy).

Versions 3.1.7 to 3.1.9

-----

\* Clint Adams <schizo@debian.org>: configuration, various fixes esp. in  
completion

\* Andrej Borsenkow <Andrej.Borsenkow@mow.siemens.ru>: design, testing and  
patches for completion system, documentation, various other fixes

\* Chmouel Boudjnah <chmouel@mandrakesoft.com>: design, testing and patches  
for completion system, documentation, various other fixes

\* Wayne Davison <wayne@clari.net>: History code

\* Alexandre Duret-Lutz <duret\_g@epita.fr>: design, testing and patches for  
completion system, documentation, various other fixes

\* Oliver Kiddle <opk@zsh.org>: AIX dynamic library support, various  
completion functions,  
web page maintenance, documentation fixes

\* Andrew Main <zefram@zsh.org>: configuration and module/DLL code, IPv6  
support, various other bits

\* Felix Rosencrantz <f\_rosencrantz@yahoo.com>: design, testing and patches  
for completion system, documentation, various other fixes

\* Bart Schaefer <schaefer@zsh.org>: documentation and many other bits,

archive maintenance

- \* Adam Spiers <adam@spiers.net>: various completion functions, source code archive maintenance, documentation fixes, prompt themes
- \* Peter Stephenson <pws@zsh.org>: pattern matching code rewrite and enhancements, typeset and local variables, zftp maintenance, general configuration, floating point support
- \* Tanaka Akira <akr@zsh.org>: CVS archive, numerous completion functions for common commands, regex shell function state machine
- \* Geoff Wing <gcw@zsh.org>: mailing list and FTP archive, various other bits
- \* Sven Wischnowsky <wischnow@zsh.org>: almost all completion code including all comp\* libraries, all core functions in completion function system; styles; job control code; rewrite of internal shell code structures and parsing and execution code to use word code; parameter code; repair roof, fix guttering, put up shelves in study, recarpet hall, ...

Other patches were submitted by Will Day, Irving Wolfe, John Williams, Vin Shelton, Akim Demaille, Helmut Jarausch, Gene Cohler, Johan Sundstrm.

Version 3.1.6

-----

- \* Sven Wischnowsky <wischnow@informatik.hu-berlin.de>: completion code, major rewrite and enhancements, including matching control, completion widgets and function system, complist module; zle code additions; job control code changes; parameters code changes; parameter module; additional autoloading code; general code changes for extra efficiency; subscribing and ordering of globbing lists.
- \* Peter Stephenson <pws@zsh.org>: zftp and mapfile modules and zf\* functions; local parameters and typeset changes; changes in autoconf system; case-independent and approximate pattern matching; various options; a few completion modules; some zle changes; FAQ.
- \* Bart Schaefer <schaefer@zsh.org>: associative array implementation; other parameter changes; keeping track of missing patches; function autoloading changes.
- \* Wayne Davison <wayne@clari.net>: History code novelties and improvements with new options.



\* Geoff Wing <gcw@zsh.org>: Zle display code fixes, mailing list and patch archive.

\* Zoltn Hidvigi <hzoli@cs.elte.hu>: AIX dynamic loading code.

\* Matt Armstrong: cygwin configuration changes.

Other improvements, bug fixes and design suggestions from all the above plus Andrej Borsenkow, Oliver Kiddle, Tanaka Akira, Naoki Wakamatsu, Tatuso Furukawa, Ville Herva, Will Day, Lehti Rahmi, Larry P. Schrof, Helmut Jarausch, Phil Pennock, Wilfredo Sanchez, Bruce Stephens, Gene Cohler, Ollivier Robert, Felix Rosencrantz.

Previous versions of zsh 3 were maintained by Zoltn Hidvigi and Andrew Main (Zefram).

Version 3.0

-----

\*

Richard Coleman <coleman@math.gatech.edu> maintained the code till the release of zsh-2.6-beta16. Converted zsh to use autoconf thus greatly improving the portability. Rewrote signal handling code. Reorganized internal hash tables and rewrote the related builtins (enable, disable, hash, unhash). Made some cleanups in exec.c.

\* Zoltn Hidvigi <hzoli@cs.elte.hu> maintained zsh from version 2.6-beta16. Rewrote most of the lexer and substitution and the related completion code. Improved sh/ksh/POSIX compatibility. Fixed lots of bugs in completion, parameter and history code. Made zsh fully 8-bit clean. Made some reorganizations in exec.c. Fixed signal handling bugs. Fixed lots of bugs in various places.

\* Peter W. Stephenson <pws@zsh.org> the maintainer of the zsh FAQ. Reorganizations in exec.c. Rewrote and reorganized the history code. Rewrote the zshcompctl manual page. Fixed several bugs related to programmable completion. Fixed several signal handling bugs. Rewrote test and read builtins. Lots of other bugfixes.

\* Andrew Main (Zefram) <zefram@zsh.org> reorganized builtin.c. Rewrote large parts of the zle vi mode. Redirection fixes. Reorganized parts of the completion code, added some enhancements and fixed lots of bugs. Made zle fully 8-bit clean. Fixed several zle bugs. Reorganized zsh option handling. Lots of other bugfixes.

\* Sven Wischnowsky <wischnow@informatik.hu-berlin.de> wrote most of the programmable completion code in zsh-2.5.0. Several bugfixes in exec.c

and jobs.c. Lots of completion bugfixes and enhancements.

\* Geoff Wing <gcw@zsh.org> rewrote most of zle\_refresh.c. Some other bugfixes.

\* Clive Messer <clive@epos.demon.co.uk> brought the texinfo documentation up-to-date.

\* Mark Borges <mdb@cdc.noaa.gov> contributed several documentation fixes including further texinfo documentation updates and maintained the zsh web pages during their tenure at www.mal.com.

\* Wayne

Davison <wayne@clari.net> improved the zle search functions and made them 8-bit clean. Some other little bugfixes.

\* Bart Schaefer <schaefer@zsh.org> submitted several bugfixes, reported lots of bugs and gave many very useful suggestions.

Version 2.5

-----

Bas de Bakker maintained zsh till the release of zsh-2.5.0. People who have contributed to zsh-2.5.0 are (in lexical order):

Chris.Moore@src.bae.co.uk (Chris Moore)  
Harald.Eikrem@delab.sintef.no  
Irving\_Wolfe@happy-man.com (Irving Wolfe)  
Jarkko.Hietaniemi@hut.fi (Jarkko Hietaniemi)  
P.Stephenson@swansea.ac.uk (Peter Stephenson)  
Richard.Sharman@software.mitel.com (Richard Sharman)  
Tero\_Kivinen@hut.FI (Tero Kivinen)  
arf@maths.nott.ac.uk (Anthony Iano-Fletcher)  
bas@phys.uva.nl (Bas de Bakker)  
benson@odi.com (Benson Margulies)  
billb@bedford.progress.com (Bill Burton)  
brown@wi.extrel.com (M. Brown)  
carlos@snfep1.if.usp.br (Carlos Carvalho)  
cedman@capitalist.princeton.edu (Carl Edman)  
chip%fin@myrddin.sybus.com (Chip Salzenberg)  
chs@apu.fi  
(Hannu Strang)  
coleman@math.gatech.edu (Richard Coleman)  
cross@eng.umd.edu (Chris Ross)  
dm@cs.brown.edu (Dimitris Michailidis)  
dmm0t@rincewind.mech.virginia.edu (David M. Meyer)  
esky@CS.UCLA.EDU (Eskandar Ensafi)  
franl@centerline.com (Fran Litterio)

gansevle@cs.utwente.nl (Fred Gansevles)  
gnohmon@ssiny.com (Ralph Betza)  
guillaum@clipper.ens.fr (Florent Guillaume)  
guthrie@math.upenn.edu (John Guthrie)  
hegedus@celeste.eng.yale.edu (Peter Hegedus)  
henryg@tusc.com.au (Henry Guillaume)  
hoh@approve.se (Goran Larsson)  
hooft@chem.ruu.nl (Rob Hooft)  
hsw1@papa.attmail.com (Stephen Harris)  
irving@happy-man.com (Irving Wolfe)  
jch@cs.cmu.edu (Jonathan Hardwick)  
jos@oce.nl (Jos Backus)  
liblit@cs.psu.edu (Benjamin Liblit)  
marc@cam.org (Marc Boucher)  
mason@werple.apana.org.au (Geoff Wing)  
mycroft@gnu.ai.mit.edu (Charles Hannum)  
mystic@axposf.pa.dec.com (D. Hall)  
norbert@i3.informatik.rwth-aachen.de (Norbert Kiesel)  
oberon@cs.tu-berlin.de (Sven Wischnowsky)  
pclink@qld.tne.oz.au (Rick)  
pem@aaii.oz.au  
(Paul Maisano)  
per@efd.lth.se (Per Foreby)  
pf@z-code.com (Paul Falstad)  
roderick@ibcinc.com (Roderick Schertler)  
schaefer@z-code.com (Bart Schaefer)  
schlangm@informatik.uni-muenchen.de (Harald Schlangmann)  
seniorr@teleport.com (Russell Senior)  
sinclair@dcs.gla.ac.uk (Duncan Sinclair)  
sterling@oldcolo.com (Bruce Sterling Woodcock)  
suzuki@otsl.oki.co.jp (Hisao Suzuki)  
tsm@cs.brown.edu (Timothy Miller)  
vogelke@c17mis.wpafb.af.mil (Karl E. Vogel)  
wacker@physik.uni-dortmund.de (Klaus Wacker)  
wacren@cis10.ens-cachan.fr (Laurent Wacrenier)  
#autoload

local expl

\_description architectures expl 'architecture'

compadd "\$@" "\$expl[@]" alpha amd64 arm64 armv7 hppa i386 landisk loongson luna88k macppc octeon sparc64  
#autoload

local expl

\_description architectures expl 'architecture'

compadd "\$@" "\$expl[@]" amd64 evbarm evbmips evbppc hpcarm i386 sparc64 xen \  
acorn32 algor alpha amiga amigappc arc atari bebox cats cesfic cobalt dreamcast \

```

emips epoc32 evbsh3 ews4800mips hp300 hppa hpcmips hpcsh ia64 ibmnws iyonix \
landisk luna68k mac68k macppc mipsco mmeye mvme68k mvmeppc netwinder news68k \
newsmips next68k ofppc pmax prep rs6000 sandpoint sbmips sgimips shark sparc \
sun2 sun3 vax x68k zaurus
#autoload

local expl

_description routing-tables expl 'routing table'
compadd "$@" "$expl[@]" - ${(s: )}${(M)}${(f)"$_call_program routing-tables netstat -R")}:# Routing tables#:
*}#*: } }
#autoload

local expl login_classes

login_classes=(${(M)}${(f)"$(</etc/login.conf)"}:#^[^#[:blank:]]*}%%[:]*})
if [[ $OSTYPE = openbsd* ]]; then
    login_classes+=(/etc/login.conf.d/*(N:t))
fi

_description login-classes expl 'login class'
compadd "$@" "$expl[@]" - $login_classes

```

## 1.78 gnutls 3.8.8-r0

### 1.78.1 Available under license :

#### Statement of Purpose

The laws of most jurisdictions throughout the world automatically confer exclusive Copyright and Related Rights (defined below) upon the creator and subsequent owner(s) (each and all, an "owner") of an original work of authorship and/or a database (each, a "Work").

Certain owners wish to permanently relinquish those rights to a Work for the purpose of contributing to a commons of creative, cultural and scientific works ("Commons") that the public can reliably and without fear of later claims of infringement build upon, modify, incorporate in other works, reuse and redistribute as freely as possible in any form whatsoever and for any purposes, including without limitation commercial purposes. These owners may contribute to the Commons to promote the ideal of a free culture and the further production of creative, cultural and scientific works, or to gain reputation or greater distribution for their Work in part through the use and efforts of others.

For these and/or

other purposes and motivations, and without any expectation of additional consideration or compensation, the person associating CC0 with a Work (the "Affirmer"), to the extent that he or she is an owner of Copyright and Related Rights in the Work, voluntarily elects to apply CC0 to the Work and publicly distribute the Work under its terms, with knowledge of his or her Copyright and Related Rights in the Work and the meaning and intended legal effect of CC0 on those rights.

1. Copyright and Related Rights. A Work made available under CC0 may be protected by copyright and related or

neighboring rights ("Copyright and Related Rights"). Copyright and Related Rights include, but are not limited to, the following:

- the right to reproduce, adapt, distribute, perform, display, communicate, and translate a Work;
- moral rights retained by the original author(s) and/or performer(s);
- publicity and privacy rights pertaining to a person's image or likeness depicted in a Work;
- rights protecting against unfair competition in regards to a Work, subject to the limitations in paragraph 4(a), below;
- rights protecting the extraction, dissemination, use and reuse of data in a Work;
- database rights (such as those arising under Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, and under any national implementation thereof, including any amended or successor version of such directive); and
- other similar, equivalent or corresponding rights throughout the world based on applicable law or treaty, and any national implementations thereof.

2. Waiver. To the greatest extent permitted by, but not in contravention of, applicable law, Affirmer hereby overtly, fully, permanently, irrevocably and unconditionally waives, abandons, and surrenders all of Affirmer's Copyright and Related Rights and associated claims and causes of action, whether now known or unknown (including existing as well as future claims

and causes of action), in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "Waiver"). Affirmer makes the Waiver for the benefit of each member of the public at large and to the detriment of Affirmer's heirs and successors, fully intending that such Waiver shall not be subject to revocation, rescission, cancellation, termination, or any other legal or equitable action to disrupt the quiet enjoyment of the Work by the public as contemplated by Affirmer's express Statement of Purpose.

3. Public License Fallback. Should any part of the Waiver for any reason be judged legally invalid or ineffective under applicable law, then the Waiver shall be preserved to the maximum extent permitted taking into account Affirmer's

express Statement of Purpose. In addition, to the extent the Waiver is so judged Affirmer hereby grants to each affected person a royalty-free, non transferable, non sublicensable, non exclusive, irrevocable and unconditional license to exercise Affirmer's Copyright and Related Rights in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "License"). The License shall be deemed effective as of the date CC0 was applied by Affirmer to the Work. Should any part of the License for any reason be judged legally invalid or ineffective under applicable law, such partial invalidity or ineffectiveness shall not invalidate the remainder of the License, and in such case Affirmer hereby affirms that he or she will

not (i) exercise any of his or her remaining Copyright and Related Rights in the Work or (ii) assert any associated claims and causes of action with respect to the Work, in either case contrary to Affirmer's express Statement of Purpose.

4. Limitations and Disclaimers.

No trademark or patent rights held by Affirmer are waived, abandoned, surrendered, licensed or otherwise affected by this document.

Affirmer offers the Work as-is and makes no representations or warranties of any kind concerning the Work,

express, implied, statutory or otherwise, including without limitation warranties of title, merchantability, fitness for a particular purpose, non infringement, or the absence of latent or other defects, accuracy, or the present or absence of errors, whether or not discoverable, all to the greatest extent permissible under applicable law.

Affirmer disclaims responsibility for clearing rights of other persons that may apply to the Work or any use thereof, including without

limitation any person's Copyright and Related Rights in the Work. Further, Affirmer disclaims responsibility for obtaining any necessary consents, permissions or other rights required for any use of the Work.

Affirmer understands and acknowledges that Creative Commons is not a party to this document and has no duty or obligation with respect to this CC0 or use of the Work.

## LICENSING

=====

Since GnuTLS version 3.1.10, the core library is released under the GNU Lesser General Public License (LGPL) version 2.1 or later (see doc/COPYING.LESSER for the license terms).

The GNU LGPL applies to the main GnuTLS library, while the included applications as well as gnutls-openssl library are under the GNU GPL version 3. The gnutls library is located in the lib/ and libdane/ directories, while the applications in src/ and, the gnutls-openssl library is at extra/.

The documentation in doc/ is under the GNU FDL license 1.3.

Note, however, that the nettle and the gmp libraries which are GnuTLS dependencies, they are distributed under a LGPLv3+ or GPLv2+ dual license. As such binaries linking to them need to adhere to either LGPLv3+ or the GPLv2+ license.

For any copyright year range specified as YYYY-ZZZZ in this package note that the range specifies every single year in that closed interval.

## GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<https://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free

software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents.

States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS

### 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the



work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

## 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The

"System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

## 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is

covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

## 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

## 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately

publish on each copy an appropriate copyright notice;  
keep intact all notices stating that this License and any  
non-permissive terms added in accord with section 7 apply to the code;  
keep intact all notices of the absence of any warranty; and give all  
recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey,  
and you may offer support or warranty protection for a fee.

#### 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to  
produce it from the Program, in the form of source code under the  
terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified  
it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is  
released under this License and any conditions added under section  
7. This requirement modifies the requirement in section  
4 to  
"keep intact all notices".
- c) You must license the entire work, as a whole, under this  
License to anyone who comes into possession of a copy. This  
License will therefore apply, along with any applicable section 7  
additional terms, to the whole of the work, and all its parts,  
regardless of how they are packaged. This License gives no  
permission to license the work in any other way, but it does not  
invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display  
Appropriate Legal Notices; however, if the Program has interactive  
interfaces that do not display Appropriate Legal Notices, your  
work need not make them do so.

A compilation of a covered work with other separate and independent  
works, which are not by their nature extensions of the covered work,  
and which are not combined with it such as to form a larger program,  
in or on a volume of a storage or distribution  
medium, is called an  
"aggregate" if the compilation and its resulting copyright are not  
used to limit the access or legal rights of the compilation's users  
beyond what the individual works permit. Inclusion of a covered work  
in an aggregate does not cause this License to apply to the other  
parts of the aggregate.

## 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no

charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

## 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

## 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright

holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

#### 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

#### 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

#### 11. Patents.



A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered

work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

#### 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

#### 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

#### 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

#### 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

#### 16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),

EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17.

Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>

This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary.

For more information on this, and how to apply and follow the GNU GPL, see  
<<https://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read  
<<https://www.gnu.org/licenses/why-not-lgpl.html>>.

The "inih" library is distributed under the New BSD license:

Copyright (c) 2009, Ben Hoyt  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Ben Hoyt nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY BEN HOYT "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE  
DISCLAIMED. IN NO EVENT SHALL BEN HOYT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS

SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.  
#####  
## Constant-time SSSE3 AES core implementation.  
## version 0.1  
##  
## By Mike Hamburg (Stanford University), 2009  
## Public domain.  
##  
## For details see [http://shiftright.org/papers/vector\\_aes/](http://shiftright.org/papers/vector_aes/) and  
## <http://crypto.stanford.edu/vpaes/>.  
#  
# \*\*\* This file is auto-generated \*\*\*  
#

GNU LESSER GENERAL PUBLIC LICENSE  
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts  
as the successor of the GNU Library Public License, version 2, hence  
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your  
freedom to share and change it. By contrast, the GNU General Public  
Licenses are intended to guarantee your freedom to share and change  
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some  
specially designated software packages--typically libraries--of the  
Free Software Foundation and other authors who  
decide to use it. You  
can use it too, but we suggest you first think carefully about whether  
this license or the ordinary General Public License is the better  
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,  
not price. Our General Public Licenses are designed to make sure that  
you have the freedom to distribute copies of free software (and charge  
for this service if you wish); that you receive source code or can get  
it if you want it; that you can change the software and use pieces of  
it in new free programs; and that you are informed that you can do  
these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

## GNU LESSER GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs



(which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public

License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify

that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative

work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the

object code for the work under the terms of Section 6.

Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany

the work with the complete corresponding

machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate

properly with a modified version of the library, if

the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials

specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally

distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies,

or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if

you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse

you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system

which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing

to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME

THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA



Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!  
Copyright (c) Meta Platforms, Inc. and affiliates.  
All rights reserved.

This source code is licensed under both the BSD-style license (found in the LICENSE file in the root directory of this source tree) and the GPLv2 (found in the COPYING file in the root directory of this source tree).  
You may select, at your option, one of the above-listed licenses.

=====  
Written by Andy Polyakov <appro@fy.chalmers.se> for the OpenSSL project. The module is, however, dual licensed under OpenSSL and CRYPTOGAMS licenses depending on where you obtain it. For further details see <https://www.openssl.org/~appro/cryptogams/>.  
=====

Copyright (c) 2006-2012, CRYPTOGAMS by <appro@openssl.org>  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain copyright notices, this list of conditions and the following disclaimer.

- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- \* Neither the name of the CRYPTOGAMS nor the names of its copyright holder and contributors may be used to endorse or promote products derived from this software without specific prior written permission.

ALTERNATIVELY, provided that this notice is retained in full, this product may be distributed under the terms of the GNU General Public License (GPL), in which case the provisions of the GPL apply INSTEAD OF those given above.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
#####  
## Constant-time SSSE3 AES core implementation.  
## version 0.1  
##  
## By Mike Hamburg (Stanford University), 2009  
## Public domain.  
##  
## For details see https://shiftleft.org/papers/vector\_aes/ and  
## https://crypto.stanford.edu/vpaes/.  
## Copyright (c) 2011-2016, Andy Polyakov <appro@openssl.org>  
## All rights reserved.  
##  
## Redistribution and use in source and binary forms, with or without  
## modification, are permitted provided that the following conditions  
## are met:  
##  
## * Redistributions of source code must retain copyright notices,  
##   this list of conditions and the following disclaimer.  
##  
## * Redistributions in binary form must reproduce the above  
##   copyright notice, this list of conditions and the following  
##   disclaimer in the documentation and/or other materials  
##   provided with the distribution.  
##  
## * Neither the name of the Andy Polyakov nor the names of its  
##   copyright holder and contributors may be used to endorse or  
##   promote products derived from this software without specific  
##   prior written permission.  
##
```

```
# ALTERNATIVELY, provided that this notice is retained in full, this
# product may be distributed under the terms of the GNU General Public
# License (GPL), in
#   which case the provisions of the GPL apply INSTEAD OF
#   those given above.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER AND CONTRIBUTORS
# "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
# A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
# OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
# SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
# LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
# DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
# THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
# (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
# OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
# *** This file is auto-generated ***
#
#
# Copyright (C) 2011-2013 Free Software Foundation, Inc.
# Copyright (C) 2013 Nikos Mavrogiannopoulos
#
# Author: Nikos Mavrogiannopoulos
#
# This file is part of GnuTLS.
#
# The GnuTLS is free software; you can redistribute it and/or
# modify it under the terms of the GNU Lesser General Public License
# as published by the Free Software Foundation; either version 2.1 of
# the License, or (at your option) any later version.
#
# This library is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
# Lesser General Public License for more details.
#
# You should have received a copy of the GNU Lesser General Public License
# along with this program. If not, see <https://www.gnu.org/licenses/>.
#
# *** This file is auto-generated ***
#
Copyright 2013 Google Inc. All Rights Reserved.

Distributed under MIT license.
See file LICENSE for detail or copy at https://opensource.org/licenses/MIT
```

Copyright (C) 1995-2022 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly      Mark Adler  
jloup@gzip.org      madler@alumni.caltech.edu  
The MIT License (MIT)

Copyright (c) 2016 Wrymouth Innovation Ltd

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.79 musl 1.2.5-r10

## 1.79.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * iconv.c
 * Implementation of SUSv4 XCU iconv utility
 * Copyright © 2011 Rich Felker
 * Licensed under the terms of the GNU General Public License, v2 or later
 */
```

Found in path(s):

\* /aports-master-main-musl-zip/aports-master-main-musl/main/musl/iconv.c

No license file was found, but licenses were detected in source scan.

```
/*_
 * Copyright (c) 1996, 1998 The NetBSD Foundation, Inc.
 * All rights reserved.
 *
 * This code is derived from software contributed to The NetBSD Foundation
 * by J.T. Conklin.
 *
 * Mostly rewritten to be used in Alpine Linux (with musl c-library)
 * by Timo Teräs.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 *
 * THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS
 * ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT
 * NOT LIMITED
 * TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS
 * BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
 * POSSIBILITY OF SUCH DAMAGE.
 */
```

Found in path(s):

\* /aports-master-main-musl-zip/aports-master-main-musl/main/musl/getconf.c

No license file was found, but licenses were detected in source scan.

/\*\_

\* Copyright (c) 2004-2006 The NetBSD Foundation, Inc.

\* All rights reserved.

\*

\* This code is derived from software contributed to The NetBSD Foundation

\* by Luke Mewburn.

\* Timo Teräs cleaned up the code for use in Alpine Linux with musl libc.

\*

\* Redistribution and use in source and binary forms, with or without

\* modification, are permitted provided that the following conditions

\* are met:

\* 1. Redistributions of source code must retain the above copyright

\* notice, this list of conditions and the following disclaimer.

\* 2. Redistributions in binary form must reproduce the above copyright

\* notice, this list of conditions and the following disclaimer in the

\* documentation and/or other materials provided with the distribution.

\*

\* THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS

\* ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED

\*

TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

\* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS

\* BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR

\* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF

\* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS

\* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN

\* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)

\* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE

\* POSSIBILITY OF SUCH DAMAGE.

\*/

Found in path(s):

\* /aports-master-main-musl-zip/aports-master-main-musl/main/musl/getent.c

## 1.80 urllib3 1.26.20

### 1.80.1 Available under license :

MIT License

Copyright (c) 2008-2020 Andrey Petrov and contributors (see CONTRIBUTORS.txt)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.81 speedtest-cli 2.1.3-r7

## 1.81.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,

including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of



this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## 1.82 py3-packaging 25.0-r0

### 1.82.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
- You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

#### END OF TERMS AND CONDITIONS

This software is made available under the terms of *\*either\** of the licenses found in LICENSE.APACHE or LICENSE.BSD. Contributions to this software is made under the terms of *\*both\** these licenses.

BSD-style without advertising clause

Copyright (c) Donald Stufft and individual contributors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,

this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.83 python-certifi 2025.01.31

### 1.83.1 Available under license :

This package contains a modified version of ca-bundle.crt:

ca-bundle.crt -- Bundle of CA Root Certificates

This is a bundle of X.509 certificates of public Certificate Authorities (CA). These were automatically extracted from Mozilla's root certificates file (certdata.txt). This file can be found in the mozilla source tree: <https://hg.mozilla.org/mozilla-central/file/tip/security/nss/lib/ckfw/builtins/certdata.txt> It contains the certificates in PEM format and therefore can be directly used with curl / libcurl / php\_curl, or with an Apache+mod\_ssl webserver for SSL client authentication. Just configure this file as the SSLCACertificateFile.#

\*\*\*\*\* BEGIN LICENSE BLOCK \*\*\*\*\*

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

\*\*\*\*\* END LICENSE BLOCK \*\*\*\*\*

@(#) \$RCSfile: certdata.txt,v \$ \$Revision: 1.80 \$ \$Date: 2011/11/03 15:11:58 \$

## 1.84 iproute 6.15.0-r0



## 1.84.1 Available under license :

### GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we

want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and

distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are

prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING

OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical  
commands ``show w'` and ``show c'` should show the appropriate  
parts of the General Public License. Of course, the commands you use may  
be called something other than ``show w'` and ``show c'`; they could even be  
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your  
school, if any, to sign a "copyright disclaimer" for the program, if  
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into  
proprietary programs. If your program is a subroutine library, you may  
consider it more useful to permit linking proprietary applications with the  
library. If this is what you want to do, use the GNU Library General  
Public License instead of this License.

## 1.85 libnftnl 1.2.9-r0

### 1.85.1 Available under license :

GNU GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your  
freedom to share and change it. By contrast, the GNU General Public  
License is intended to guarantee your freedom to share and change free  
software--to make sure the software is free for all its users. This  
General Public License applies to most of the Free Software  
Foundation's software and to any other program whose authors commit to  
using it. (Some other Free Software Foundation software is covered by  
the GNU Library General Public License instead.) You can apply it to  
your programs, too.

When we speak of free software, we are referring to freedom, not



price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law:

that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any

associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you

may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands  
`show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General

Public License instead of this License.

# 1.86 e2fsprogs 1.47.2-r2

## 1.86.1 Available under license :

Index: tdbsa/tdb.c

=====

--- tdbsa.orig/tdb.c

+++ tdbsa/tdb.c

@@ -4,11 +4,11 @@ Rev: 23371

Last Changed Date: 2007-06-06 20:14:06 -0400 (Wed, 06 Jun 2007)

\*/

/\*

- Unix SMB/CIFS implementation.
- + trivial database library - standalone version

- trivial database library - private includes

-

- Copyright (C) Andrew Tridgell 2005
- + Copyright (C) Andrew Tridgell 1999-2005
- + Copyright (C) Jeremy Allison 2000-2006
- + Copyright (C) Paul 'Rusty' Russell 2000

\*\* NOTE! The following LGPL license applies to the tdb

\*\* library. This does NOT imply that all of Samba is released

EXT2ED is hereby placed under the terms of the GNU General Public License.

Follows the GNU license.

Gadi Oxman, August 1995

-----

### GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

675 Mass Ave, Cambridge, MA 02139, USA

Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your  
freedom to share and change it. By contrast, the GNU General Public  
License is intended to guarantee your freedom to share and change free  
software--to make sure the software is free for all its users. This  
General Public License applies to most of the Free Software



Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a

notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you

received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access

to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent

infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program

specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY

TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest

to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY;  
for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into

proprietary programs. If your program is a subroutine library, you may consider

it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

This package, the EXT2 filesystem utilities, are made available under the GNU Public License version 2, with the exception of the lib/ext2fs and lib/e2p libraries, which are made available under the GNU Library General Public License Version 2, the lib/uuid library which is made available under a BSD-style license and the lib/et and lib/ss libraries which are made available under an MIT-style license. Please see lib/uuid/COPYING for more details for the license for the files comprising the libuuid library, and the source file headers of the libet and libss libraries for more information.

The most recent officially distributed version can be found at <http://e2fsprogs.sourceforge.net>. If you need to make a distribution, that's the one you should use. If there is some reason why you'd like a more recent version that is still in ALPHA testing (i.e., either using the "WIP" test distributions or one from the hg or git repository from the development branch, please contact me ([tytso@mit.edu](mailto:tytso@mit.edu))

before you ship. The release schedules for this package are flexible, if you give me enough lead time.

Theodore Ts'o  
23-June-2007

---

## GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software



is covered by  
the GNU Library General Public License instead.) You can apply it to  
your programs, too.

When we speak of free software, we are referring to freedom, not  
price. Our General Public Licenses are designed to make sure that you  
have the freedom to distribute copies of free software (and charge for  
this service if you wish), that you receive source code or can get it  
if you want it, that you can change the software or use pieces of it  
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid  
anyone to deny you these rights or to ask you to surrender the rights.  
These restrictions translate to certain responsibilities for you if you  
distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether  
gratis or for a fee, you must give the recipients all the rights that  
you have. You must make sure that they, too, receive or can get the  
source  
code. And you must show them these terms so they know their  
rights.

We protect your rights with two steps: (1) copyright the software, and  
(2) offer you this license which gives you legal permission to copy,  
distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain  
that everyone understands that there is no warranty for this free  
software. If the software is modified by someone else and passed on, we  
want its recipients to know that what they have is not the original, so  
that any problems introduced by others will not reflect on the original  
authors' reputations.

Finally, any free program is threatened constantly by software  
patents. We wish to avoid the danger that redistributors of a free  
program will individually obtain patent licenses, in effect making the  
program proprietary. To prevent this, we have made it clear that any  
patent must be licensed for everyone's free use or not licensed at all.

The precise terms  
and conditions for copying, distribution and  
modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains

a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered

only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this

License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not

excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free

Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name  
of author>

This program is free software; you can redistribute it and/or modify  
it under the terms of the GNU General Public License as published by  
the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.

You should have received a copy of the GNU General Public License  
along with this program; if not, write to the Free Software  
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this  
when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision
comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate  
parts of the General Public License. Of course, the commands you use may  
be called something other than `show w' and `show c'; they could even be  
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your  
school, if any, to sign a "copyright disclaimer" for the program, if  
necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into  
proprietary programs. If your program is a

subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

---

## GNU LIBRARY GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide



complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to

permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

## GNU LIBRARY GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

#### Activities

other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of

running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the

source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work

during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
- b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library

facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this

License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.



14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

##### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the

"copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!

#

# This is a Makefile stub which handles the creation of BSD shared  
# libraries.

#

# In order to use this stub, the following makefile variables must be defined.

#

# BSDLIB\_VERSION = 1.0

# BSDLIB\_IMAGE = libce

# BSDLIB\_MYDIR = et

# BSDLIB\_INSTALL\_DIR = \$(SHLIBDIR)

#

all:: image

real-subdirs:: Makefile

```
@echo " MKDIR pic"
```

```
@mkdir -p pic
```

```
BSD_LIB = $(BSDLIB_IMAGE).so.$(BSDLIB_VERSION)
```

```
BSDLIB_PIC_FLAG = -fpic
```

```
image: $(BSD_LIB)
```

```
$(BSD_LIB): $(OBJS)
```

```
(cd pic; ld -Bshareable -o $(BSD_LIB) $(LDFLAGS_SHLIB) $(OBJS))
```

```
$(MV) pic/$(BSD_LIB) .
```

```
$(RM) -f ../$(BSD_LIB)
```

```
(cd ..; $(LN) $(LINK_BUILD_FLAGS) \
```

```
`echo $(my_dir) | sed -e 's;lib/;;'/'$(BSD_LIB) $(BSD_LIB))
```

```
install-shlibs install:: $(BSD_LIB)
```

```
@echo " INSTALL_PROGRAM $(BSDLIB_INSTALL_DIR)/$(BSD_LIB)"
```

```
@$(INSTALL_PROGRAM) $(BSD_LIB) \
```

```
$(DESTDIR)$ (BSDLIB_INSTALL_DIR)/$(BSD_LIB)
```

```
@-$(LDCONFIG)
```

```
install-strip: install
```

```
install-shlibs-strip:: install-shlibs
```

```
uninstall-shlibs uninstall::
```

```
$(RM) -f $(DESTDIR)$ (BSDLIB_INSTALL_DIR)/$(BSD_LIB)
```

```
clean::
```

```
$(RM)
```

```
-rf pic
```

```
$(RM) -f $(BSD_LIB)
```

```
$(RM) -f ../$(BSD_LIB)
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, and the entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ALL OF WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF NOT ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Format: <https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Upstream-Name: e2fsprogs

Upstream-Contact: Theodore Ts'o <tytso@mit.edu>

Source: <http://sourceforge.net/projects/e2fsprogs>

Comment:

This is the Debian GNU/Linux prepackaged version of the EXT2 file system utilities (e2fsck, mke2fs, etc.). The EXT2 utilities were written by Theodore Ts'o <tytso@mit.edu> and Remy Card <card@masi.ibp.fr>.

Files: \*

Copyright: 1993-2022 Theodore Ts'o <tytso@mit.edu>

License: GPL-2

Comment: Some files are GPL-2+ licensed or do not carry any license version.

They are distributable as GPL-2 and listed as such for simplification.

Files: debian/\*

Copyright: 2003-2007 Theodore Ts'o <tytso@mit.edu>

1997-2003 Yann Dirson <dirson@debian.org>

2001 Alcove <<http://www.alcove.com/>>

1997 Klee Dienes

1995-1996 Michael Nonweiler <mrn20@cam.ac.uk>

License: GPL-2

Files: debian/fuse-ext2

Copyright: 2008-2010 Alper Akcan <alper.akcan@gmail.com>

2009-2010 Renzo Davoli <renzo@cs.unibo.it>

License: GPL-2+

Files: debian/fuse-ext2.1

Copyright: 2024 <nabijaczleweli@nabijaczleweli.xyz>

License: 0BSD

Files: lib/et/\*

lib/ss/\*

Copyright: 1987-1988 MIT Student Information Processing Board

2003 Theodore Ts'o

License: Expat

Permission to use, copy, modify, and distribute this software and its documentation for any purpose is hereby granted, provided that the names of M.I.T. and the M.I.T. S.I.P.B. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. M.I.T. and the M.I.T. S.I.P.B. make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Files: lib/et/vfprintf.c

Copyright: 1988 Regents of the University of California

License: BSD-3-Clause-Variant

Redistribution and use in source and binary forms are permitted provided that the above copyright

notice and this paragraph are

duplicated in all such forms and that any documentation,

advertising materials, and other materials related to such

distribution and use acknowledge that the software was developed

by the University of California, Berkeley. The name of the

University may not be used to endorse or promote products derived

from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Files: lib/et/com\_right.c

Comment: Taken from lib/com\_err/error.c from Kerberos4kth distribution.

Copyright: (c) 2003 by Theodore Ts'o

Copyright (c) 1997, 1998, 2001 Kungliga Tekniska Hgskolan

(Royal Institute of Technology, Stockholm, Sweden).

All rights reserved.

License: BSD-3-Clause

Files: lib/et/test\_cases/imap\_err.et

Copyright: 1994-2008 Carnegie Mellon University.

License: BSD-4-Clause-CMU

Files:

lib/ext2fs/\*

lib/e2p/\*

lib/support/print\_fs\_flags.c

misc/e2fuzz.c

Copyright: 1995-1999, 2002, 2004-2006 Theodore Ts'o <tytso@mit.edu>

1992-1994 Remy Card <card@masi.ibp.fr>

Laboratoire MASI, Institut Blaise Pascal

Universite Pierre et Marie Curie (Paris VI)

2001 Andreas Gruenbacher, <a.gruenbacher@computer.org>

2007 IBM Corporation

2010 Red Hat, Inc., Lukas Czerner <lczerner@redhat.com>  
2014 Oracle  
2012 Zheng Liu <wenqing.lz@taobao.com>  
2012 Intel Corporation  
2001 Daniel Phillips  
2002 Theodore Ts'o.  
2002 Andreas Dilger  
2006 Cluster File Systems, Inc.  
2006, 2007 by Andreas Dilger <adilger@clusterfs.com>  
2003,2004 Cluster File Systems, Inc <info@clusterfs.com>  
1995 Theodore Ts'o.  
2003 VMware, Inc.  
2015 Jan Kara  
2018 Collabora

Ltd.

1997 Theodore Ts'o  
1997 Mark Habersack  
2011 Whamcloud, Inc.  
2004 Sam Hocevar <sam@hocevar.net>  
2014 Google Inc.  
1993, 1994, 1995 Theodore Ts'o.  
1998 Andrey Shedel <andreys@ns.cr.cyco.com>

License: LGPL-2

Comment: Some files in lib/ext2fs claim

"This file may be redistributed under the terms of the GNU Public License."  
and NOTICE clarifies them to be licensed under LGPL-2.

Files: lib/ext2fs/tdb\*

Copyright: 1999-2005 Andrew Tridgell

2000-2006 Jeremy Allison

2000 Paul 'Rusty' Russell

License: LGPL-2

Files: lib/ext2fs/tdbtool.c

Copyright: 1999-2000 Andrew Tridgell

2000 Paul 'Rusty' Russell

2000 Jeremy Allison

2001 Andrew Esh

License: GPL-2

Files: lib/ext2fs/kernel-jbd.h

Copyright: 1998-2000 Red Hat, Inc

License: GPL-2

Files: lib/ext2fs/nls\_utf8.c

lib/ext2fs/utf8n.h

util/mkutf8data.c

Copyright: 2014 SGI

2018 Collabora Ltd.  
License:  
GPL-2

Files: lib/ext2fs/crc16.\*  
Copyright: 2005 Ben Gardner <bgardner@wabtec.com>  
License: GPL-2

Files: lib/ext2fs/crc32c\*  
Copyright:  
August 26, 2011 Darrick J. Wong <djwong at us.ibm.com>  
Reuse Bob Pearson's slice-by-8 implementation for e2fsprogs.  
.  
July 20, 2011 Bob Pearson <rpearson at systemfabricworks.com>  
...  
Oct 15, 2000 Matt Domsch <Matt\_Domsch@dell.com>  
...  
Oct 12, 2000 Matt Domsch <Matt\_Domsch@dell.com>  
License: GPL-2

Files: lib/uuid/\*  
Copyright: 1996-1999, 2007 Theodore Ts'o.  
License: BSD-3-Clause

Files: lib/uuid/\*.3.in  
Copyright: 1999 Andreas Dilger <adilger@enel.ualgary.ca>  
License: BSD-3-Clause

Files: contrib/fallocate.c  
Copyright: 2008 Red Hat, Inc. All rights reserved.  
2003-2005 Silicon Graphics, Inc.  
License: GPL-2

Files: contrib/dconf  
Copyright: 1994 Ulrich Windl <Ulrich.Windl@rz.uni-regensburg.de>  
License: GPL-2

Files: CleanSpec.mk  
contrib/android/ext2simg.c  
Copyright: 2007, 2016 The Android Open Source Project  
License:  
Apache-2

Files: e2fsck/\*  
debugfs/do\_journal.c  
debugfs/journal.\*  
debugfs/logdump.c  
debugfs/xattrs.c

lib/support/plausible.c

misc/fuse2fs.c

misc/fsmap.h

Copyright: 1993-2011 Theodore Ts'o

1999-2000 Red Hat Software

2000 Red Hat corp

2001 Red Hat, Inc.

2000 Stephen C. Tweedie

2000 Andreas Dilger

2014 Oracle

2019 Google LLC

License: GPL-2

Files: e2fsck/mtrace.\*

include/nonunix/getopt.h

Copyright: 1985-2015 Free Software Foundation, Inc.

License: LGPL-2

Files: ext2ed/\*

Copyright: 1995 Gadi Oxman

License: GPL-2

Files: include/mingw/unistd.h

Copyright: 2004-2006 Kern Sibbald

2014 Adam Kropelin

License: GPL-2

Files: lib/support/devname.\*

Copyright: 2010 Red Hat, Inc., Lukas Czermer <lczermer@redhat.com>

License: GPL-2

Files: lib/blkid/\*

misc/blkid.c

Copyright: 1999, 2001 by Andries Brouwer

1994-1997, 1999-2004 Theodore Ts'o <tytso@mit.edu>

2001 Andreas Dilger

License: LGPL-2

Files: misc/blkid.8.in

misc/uuidgen.1.in

Copyright: 1999, 2000 Andreas Dilger <adilger@turbolinux.com>

License: GPL-2

Files: misc/create\_inode.c

Copyright: 2014 Robert Yang <liezhi.yang@windriver.com>

License: LGPL-2

Files: misc/e2freefrag.c



Copyright: 2009 Sun Microsystems, Inc.

License: GPL-2

Files: misc/e2undo.c

Copyright: 2007 IBM Corporation

License: GPL-2

Files: misc/e4defrag.c

Copyright: Akira Fujita <a-fujita@rs.jp.nec.com>

Takashi Sato <t-sato@yk.jp.nec.com>

License: GPL-2

Files: misc/uuidgen.c

Copyright: 1999 Andreas Dilger and Theodore Ts'o

License: GPL-2

Files: misc/e2fuzz.sh

Copyright: 2014 Oracle

License: GPL-2

Files: misc/e4crypt.c

Copyright: 2014 Google, Inc.

License: GPL-2

Files: ext2ed/ext2.descriptors

misc/badblocks.c

misc/chattr.c

misc/dumpe2fs.c

misc/lsattr.c

misc/mklost+found.c

misc/tune2fs.c

Copyright: 1995-1999 by Theodore Ts'o

1992-1994 Remy Card <card@masi.ibp.fr>

Laboratoire MASI, Institut Blaise Pascal

Universite Pierre et Marie Curie (Paris VI)

1999 by David Beattie

Linus Torvalds <Linus.Torvalds@cs.helsinki.fi>

License: GPL-2

Files: po/fr.po

po/vi.po

Copyright: 2008, 2014 Theodore Ts'o

1996, 2014 Free Software Foundation, Inc.

License: GPL-2

Files: resize/\*

Copyright: 1997-2000 by Theodore Ts'o and PowerQuest, Inc.

License: GPL-2

Files: scrub/\*

Copyright: 2018 Oracle

License: GPL-2

Files: tests/fuzz/\*.cc

Copyright: 2020 Google LLC

License: Apache-2

Files: lib/support/argv\_parse.\*

Copyright: 1999 Theodore Ts'o.

License: ISC

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies. THE SOFTWARE IS PROVIDED "AS IS" AND THEODORE TS'O (THE AUTHOR) DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Files: lib/support/profile.\*

Copyright: 2005, 2006 by Theodore Ts'o.

1985-2005 by the Massachusetts Institute of Technology.

All rights reserved.

License: GPL or MIT-US-export

This file may be redistributed under the terms of the GNU Public License.

.

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of M.I.T. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Furthermore if you modify this software you must label

your software as modified software and not distribute it in such a fashion that it might be confused with the original MIT software. M.I.T. makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

.  
THIS SOFTWARE IS PROVIDED ``AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Comment:

On Debian

systems, the complete text of the GNU General Public License can be found in "/usr/share/common-licenses/GPL".

Files: lib/support/dict.\*

Copyright: 1997 Kaz Kylheku <kaz@ashi.footprints.net>

License: Kazlib

All rights are reserved by the author, with the following exceptions:

Permission is granted to freely reproduce and distribute this software, possibly in exchange for a fee, provided that this copyright notice appears intact. Permission is also granted to adapt this software to produce derivative works, as long as the modified versions carry this copyright notice and additional notices stating that the work has been modified. This source code may be translated into executable form and incorporated into proprietary software; there is no requirement for such software to contain a copyright notice related to this source.

Comment: The work has been modified.

Files: doc/libext2fs.texinfo

Copyright: 1997-2014 Theodore Ts'o

License: Latex2e

Permission is granted to process this file through TeX and print the results, provided the printed document carries copying permission notice identical to this one except for the removal of this paragraph (this paragraph not being relevant to the printed manual).

.  
Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

.  
Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the author.

Files: doc/texinfo.tex

lib/et/texinfo.tex

Copyright: 1985-2006 Free Software Foundation, Inc.

License: GPL-2+ with Texinfo exception

This texinfo.tex file is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as

published by the Free Software Foundation; either version 2, or (at your option) any later version.

.

This texinfo.tex file is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

.

You should have received a copy of the GNU General Public License along with this texinfo.tex file; see the file COPYING. If not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

.

As a special exception, when this file is read by TeX when processing a Texinfo source document, you may use the result without restriction. (This has been our intent since Texinfo was invented.)

License: Apache-2

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

.

<http://www.apache.org/licenses/LICENSE-2.0>

.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Comment:

On Debian systems, the complete text of the Apache License, Version 2.0 can be found in `"/usr/share/common-licenses/Apache-2.0"`.

License: BSD-3-Clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, and the entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the

documentation

and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ALL OF WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF NOT ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-4-Clause-CMU

Redistribution and use in source

and binary forms, with or without

modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name "Carnegie Mellon University" must not be used to endorse or promote products derived from this software without prior written permission. For permission or any legal details, please contact  
Carnegie Mellon University  
Center for Technology Transfer and Enterprise Creation  
4615 Forbes Avenue  
Suite 302  
Pittsburgh, PA 15213  
(412) 268-7393, fax: (412) 268-7395  
innovation@andrew.cmu.edu
4. Redistributions of any form whatsoever must retain the following

acknowledgment:

"This product includes software developed by Computing Services  
at Carnegie Mellon University (<http://www.cmu.edu/computing/>)."

.  
CARNEGIE MELLON UNIVERSITY DISCLAIMS ALL WARRANTIES WITH REGARD TO  
THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY  
AND FITNESS, IN NO EVENT SHALL CARNEGIE MELLON UNIVERSITY BE LIABLE  
FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES  
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN  
AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING  
OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

License: GPL-2

This program is free software; you can redistribute it and/or modify  
it under the terms version 2 of the GNU General Public License  
as published by the Free Software Foundation.

.  
This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A  
PARTICULAR PURPOSE. See the  
GNU General Public License for more details.

.  
You should have received a copy of the GNU General Public License  
along with this program. If not, see <https://www.gnu.org/licenses/>.

Comment:

On Debian systems, the complete text of the GNU General  
Public License version 2 can be found in "/usr/share/common-licenses/GPL-2".

License: GPL-2+

This program is free software; you can redistribute it and/or modify  
it under the terms of the GNU General Public License as published by  
the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.

.  
This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.

.  
You should have received a copy of the GNU General Public License  
along with this program (in the main directory of the fuse-ext2  
distribution

in the file COPYING); if not, write to the Free Software  
Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA.

Comment:

On Debian systems, the complete text of the GNU General  
Public License version 2 can be found in "/usr/share/common-licenses/GPL-2".

License: 0BSD

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted.

.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

License: LGPL-2

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; version 2 of the License.

.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

.

You should have received a copy of the GNU Library General Public License along with this library. If not, see <<https://www.gnu.org/licenses/>>.

Comment:

On Debian systems, the complete text of the GNU Library General Public License version 2 can be found in "/usr/share/common-licenses/LGPL-2".

/\*

Copyright 1987 by the Student Information Processing Board  
of the Massachusetts Institute of Technology

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of M.I.T. and the M.I.T. S.I.P.B. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. M.I.T. and the M.I.T. S.I.P.B. make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

\*/

# 1.87 libnfnetwork 1.0.2-r3

## 1.87.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

675 Mass Ave, Cambridge, MA 02139, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.



Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole

or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work

based on it,  
under Section 2) in object code or executable form under the terms of  
Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6.

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING

WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),

EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute it under certain conditions; type ``show c'` for details.

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program ``Gnomovision'` (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

## 1.88 platformdirs 4.2.2

### 1.88.1 Available under license :

MIT License

Copyright (c) 2010-202x The platformdirs developers

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT

OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.89 libuv 1.51.0-r0

### 1.89.1 Available under license :

Attribution 4.0 International

=====

Creative Commons Corporation ("Creative Commons") is not a law firm and does not provide legal services or legal advice. Distribution of Creative Commons public licenses does not create a lawyer-client or other relationship. Creative Commons makes its licenses and related information available on an "as-is" basis. Creative Commons gives no warranties regarding its licenses, any material licensed under their terms and conditions, or any related information. Creative Commons disclaims all liability for damages resulting from their use to the fullest extent possible.

#### Using Creative Commons Public Licenses

Creative Commons public licenses provide a standard set of terms and conditions that creators and other rights holders may use to share original works of authorship and other material subject to copyright and certain other rights specified in the public license below. The following considerations are for informational purposes only, are not exhaustive, and do not form part of our licenses.

Considerations for licensors: Our public licenses are intended for use by those authorized to give the public permission to use material in ways otherwise restricted by copyright and certain other rights. Our licenses are irrevocable. Licensors should read and understand the terms and conditions of the license they choose before applying it. Licensors should also secure all rights necessary before applying our licenses so that the public can reuse the material as expected. Licensors should clearly mark any material not subject to the license. This includes other CC-licensed material, or material used under an exception or limitation to copyright. More considerations for licensors: [wiki.creativecommons.org/Considerations\\_for\\_licensors](http://wiki.creativecommons.org/Considerations_for_licensors)

Considerations for the public: By using one of our public licenses, a licensor



grants the public permission to use the licensed material under specified terms and conditions. If the licensor's permission is not necessary for any reason--for example, because of any applicable exception or limitation to copyright--then that use is not regulated by the license. Our licenses grant only permissions under copyright and certain other rights that a licensor has authority to grant. Use of the licensed material may still be restricted for other reasons, including because others have copyright or other rights in the material. A licensor may make special requests, such as asking that all changes be marked or described. Although not required by our licenses, you are encouraged to respect those requests where reasonable. More considerations for the public:

[wiki.creativecommons.org/Considerations\\_for\\_licensees](http://wiki.creativecommons.org/Considerations_for_licensees)

=====

## Creative Commons Attribution 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

### Section 1 -- Definitions.

- a. Adapted Material means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.
- b. Adapter's License means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.

- c. Copyright and Similar Rights means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.
- d. Effective Technological Measures means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.
- e. Exceptions and Limitations means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.
- f. Licensed Material means the artistic or literary work, database, or other material to which the Licensor applied this Public License.
- g. Licensed Rights means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.
- h. Licensor means the individual(s) or entity(ies) granting rights under this Public License.
- i. Share means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.
- j. Sui Generis Database Rights means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.
- k. You means the individual or entity exercising the Licensed Rights under this Public License. Your has a corresponding meaning.

## Section 2 -- Scope.

### a. License grant.

1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable

license to

exercise the Licensed Rights in the Licensed Material to:

- a. reproduce and Share the Licensed Material, in whole or in part; and

- b. produce, reproduce, and Share Adapted Material.

2. Exceptions and Limitations. For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.

3. Term. The term of this Public License is specified in Section 6(a).

4. Media and formats; technical modifications allowed. The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section 2(a) (4) never produces Adapted Material.

5. Downstream recipients.

- a. Offer from the Licensor -- Licensed Material. Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.

- b. No downstream restrictions. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the

Licensed Rights by any recipient of the Licensed

Material.

6. No endorsement. Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i).

b. Other rights.

1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.

2. Patent and trademark rights are not licensed under this Public License.

3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties.

### Section 3 -- License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

a. Attribution.

1. If You Share the Licensed Material (including in modified form), You must:

a. retain the following if it is supplied by the Licensor with the Licensed Material:

i. identification of the creator(s) of the Licensed Material and any others designated to receive

attribution, in any reasonable manner requested by the Licensor (including by pseudonym if

designated);

ii. a copyright notice;

iii. a notice that refers to this Public License;

iv. a notice that refers to the disclaimer of warranties;

v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;

b. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and

c. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.

2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.

3. If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.

4. If You Share Adapted Material You produce, the Adapter's License You apply must not prevent recipients of the Adapted Material from complying with this Public License.

#### Section 4 -- Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database;

- b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material; and
- c. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

#### Section 5 -- Disclaimer of Warranties and Limitation of Liability.

- a. UNLESS OTHERWISE SEPARATELY UNDERTAKEN BY THE LICENSOR, TO THE EXTENT POSSIBLE, THE LICENSOR OFFERS THE LICENSED MATERIAL AS-IS AND AS-AVAILABLE, AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE LICENSED MATERIAL, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHER. THIS INCLUDES, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OR ABSENCE OF ERRORS, WHETHER OR NOT KNOWN OR DISCOVERABLE. WHERE DISCLAIMERS OF WARRANTIES ARE NOT ALLOWED IN FULL OR IN PART, THIS DISCLAIMER MAY NOT APPLY TO YOU.
- b. TO THE EXTENT POSSIBLE, IN NO EVENT WILL THE LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE) OR OTHERWISE FOR ANY DIRECT, SPECIAL, INDIRECT, INCIDENTAL, CONSEQUENTIAL, PUNITIVE, EXEMPLARY, OR OTHER LOSSES, COSTS, EXPENSES, OR DAMAGES ARISING OUT OF THIS PUBLIC LICENSE OR USE OF THE LICENSED MATERIAL, EVEN IF THE LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSSES, COSTS, EXPENSES, OR DAMAGES. WHERE A LIMITATION OF LIABILITY IS NOT ALLOWED IN FULL OR IN PART, THIS LIMITATION MAY NOT APPLY TO YOU.
- c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

#### Section 6 -- Term and Termination.

- a. This Public License

applies for the term of the Copyright and

Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.

b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:

1. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or
2. upon express reinstatement by the Licensor.

For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.

d. Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

#### Section 7 -- Other Terms and Conditions.

- a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.
- b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

#### Section 8 -- Interpretation.

- a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully be made without permission under this Public License.
- b. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision

cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions.

c. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.

d. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.

=====

Creative Commons is not a party to its public licenses. Notwithstanding, Creative Commons may elect to apply one of its public licenses to material it publishes and in those instances will be considered the Licensor. The text of the Creative Commons public licenses is dedicated to the public domain under the CC0 Public Domain Dedication. Except for the limited purpose of indicating that material is shared under a Creative Commons public license or as otherwise permitted by the Creative Commons policies published at [creativecommons.org/policies](http://creativecommons.org/policies), Creative Commons does not authorize the use of the trademark "Creative Commons" or any other trademark or logo of Creative Commons without its prior written consent including, without limitation, in connection with any unauthorized modifications to any of its public licenses or any other arrangements, understandings, or agreements concerning use of licensed material. For the avoidance of doubt, this paragraph does not form part of the public licenses.

Creative Commons may be contacted at [creativecommons.org](http://creativecommons.org).  
Copyright (c) 2015-present libuv project contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR



IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR

IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS  
IN THE SOFTWARE.

/\* Copyright Joyent, Inc. and other Node contributors. All rights reserved.

\*

\* Permission is hereby granted, free of charge, to any person obtaining a copy  
\* of this software and associated documentation files (the "Software"), to  
\* deal in the Software without restriction, including without limitation the  
\* rights to use, copy, modify, merge, publish, distribute, sublicense, and/or  
\* sell copies of the Software, and to permit persons to whom the Software is  
\* furnished to do so, subject to the following conditions:

\*

\* The above copyright notice and this permission notice shall be included in  
\* all copies or substantial portions of the Software.

\*

\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
\* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
\* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE  
\* AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  
\* LIABILITY, WHETHER IN  
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING  
\* FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS  
\* IN THE SOFTWARE.

\*/

#ifndef UV\_BSD\_H

#define UV\_BSD\_H

#define UV\_PLATFORM\_FS\_EVENT\_FIELDS \

uv\_\_io\_t event\_watcher; \

#define UV\_IO\_PRIVATE\_PLATFORM\_FIELDS \

int rcount; \

int wcount; \

#define UV\_HAVE\_KQUEUE 1

#endif /\* UV\_BSD\_H \*/

This license applies to parts of libuv originating from the  
<https://github.com/joyent/libuv> repository:

====

Copyright Joyent, Inc. and other Node contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

This license applies to all parts of libuv that are not externally maintained libraries.

The externally maintained libraries used by libuv are:

- tree.h (from FreeBSD), copyright Niels Provos. Two clause BSD license.
- inet\_pton and inet\_ntop implementations, contained in src/inet.c, are copyright the Internet Systems Consortium, Inc., and licensed under the ISC license.

## 1.90 kerberos 1.21.3-r0

### 1.90.1 Available under license :

[ NOTE: MIT has only incorporated the mechglue and spnego change, and not the incremental propagation changes. The filenames are different between the Sun and MIT sources. The actual MIT filenames appear in the top-level README file. Original text of Sun's LICENSE file follows. ]

Subject to the license set forth below, Sun Microsystems, Inc. donates the attached files to MIT for the purpose of including these modifications and additions in future versions of the Kerberos system.

Many of the files attached are subject to licenses issued by other entities, including OpenVision, MIT, and FundsXpress. See the

individual files, and/or related Readme files, for these licenses.

In addition Sun requires that the license set forth below be incorporated into any future version of the Kerberos system which contains portions of the files attached. The following files must be listed, in the top level Readme file, as being provided subject to such license:

```
cmd/krb5/iprop/iprop.x
cmd/krb5/iprop/iprop_hdr.h
cmd/krb5/kadmin/server/ipropd_svc.c
cmd/krb5/kproplog/kproplog.c
cmd/krb5/slave/kpropd_rpc.c
lib/gss_mechs/mech_krb5/et/kdb5_err.c
lib/gss_mechs/mech_spnego/mech/gssapiP_spnego.h
lib/gss_mechs/mech_spnego/mech/spnego_mech.c
lib/krb5/kadm5/kadm_host_srv_names.c
lib/krb5/kdb/kdb_convert.c
lib/krb5/kdb/kdb_hdr.h
lib/krb5/kdb/kdb_log.c
lib/krb5/kdb/kdb_log.h
lib/libgss/g_accept_sec_context.c
lib/libgss/g_acquire_cred.c
lib/libgss/g_canon_name.c
lib/libgss/g_compare_name.c
lib/libgss/g_context_time.c
lib/libgss/g_delete_sec_context.c
lib/libgss/g_dsp_name.c
lib/libgss/g_dsp_status.c
lib/libgss/g_dup_name.c
lib/libgss/g_exp_sec_context.c
lib/libgss/g_export_name.c
lib/libgss/g_glue.c
lib/libgss/g_imp_name.c
lib/libgss/g_imp_sec_context.c
lib/libgss/g_init_sec_context.c
lib/libgss/g_initialize.c
lib/libgss/g_inquire_context.c
lib/libgss/g_inquire_cred.c
lib/libgss/g_inquire_names.c
lib/libgss/g_process_context.c
lib/libgss/g_rel_buffer.c
lib/libgss/g_rel_cred.c
lib/libgss/g_rel_name.c
lib/libgss/g_rel_oid_set.c
lib/libgss/g_seal.c
lib/libgss/g_sign.c
lib/libgss/g_store_cred.c
```

lib/libgss/g\_unseal.c  
lib/libgss/g\_userok.c  
lib/libgss/g\_utils.c  
lib/libgss/g\_verify.c  
lib/libgss/gssd\_pname\_to\_uid.c  
uts/common/gssapi/include/gssapi\_err\_generic.h  
uts/common/gssapi/include/mechglueP.h

Sun's

License is as follows:

Copyright (c) 2004 Sun Microsystems, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright

=====

Copyright |copy| 1985-2024 by the Massachusetts Institute of Technology and its contributors. All rights reserved.

See :ref:`mitK5license` for additional copyright and license information.

Copyright (C) 1985-2024 by the Massachusetts Institute of Technology.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

\* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Downloading of this software may constitute an export of cryptographic software from the United States of America that is subject to the United States Export Administration Regulations (EAR), 15 CFR 730-774. Additional laws or regulations may apply. It is the responsibility of the person or entity contemplating export to comply with all applicable export laws and regulations, including obtaining any required license from the U.S. government.

The U.S. government prohibits export of encryption source code to certain countries and individuals, including, but not limited to, the countries of Cuba, Iran, North Korea, Sudan, Syria, and residents and nationals of those countries.

Documentation components of this software distribution are licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License. (<https://creativecommons.org/licenses/by-sa/3.0/>)

Individual source code files are copyright MIT, Cygnus Support, Novell, OpenVision Technologies, Oracle, Red Hat, Sun Microsystems, FundsXpress, and others.

Project Athena, Athena, Athena MUSE, Discuss, Hesiod, Kerberos, Moira, and Zephyr are trademarks of the Massachusetts Institute of Technology (MIT). No commercial use of these trademarks may be made without prior written permission of MIT.

"Commercial use" means use of a name in a product or other for-profit

manner. It does NOT prevent a commercial firm from referring to the MIT trademarks in order to convey information (although in doing so, recognition of their trademark status should be given).

=====

The following copyright and permission notice applies to the OpenVision Kerberos Administration system located in "kadmin/create", "kadmin/dbutil", "kadmin/passwd", "kadmin/server", "lib/kadm5", and portions of "lib/rpc":

Copyright, OpenVision Technologies, Inc., 1993-1996, All Rights Reserved

WARNING: Retrieving the OpenVision Kerberos Administration system source code, as described below, indicates your acceptance of the following terms. If you do not agree to the following terms, do not retrieve the OpenVision Kerberos administration system.

You may freely use and distribute the Source Code and Object Code compiled from it, with or without modification, but this Source Code is provided to you "AS IS" EXCLUSIVE OF ANY WARRANTY, INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, OR ANY OTHER WARRANTY, WHETHER EXPRESS OR IMPLIED. IN NO EVENT WILL OPENVISION HAVE ANY LIABILITY FOR ANY LOST PROFITS, LOSS OF DATA OR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, INCLUDING, WITHOUT LIMITATION, THOSE RESULTING FROM THE USE OF THE SOURCE CODE, OR THE FAILURE OF THE SOURCE CODE TO PERFORM, OR FOR ANY OTHER REASON.

OpenVision retains all copyrights in the donated Source Code. OpenVision also retains copyright to derivative works of the Source Code, whether created by OpenVision or by a third party. The OpenVision copyright notice must be preserved if derivative works are made based on the donated Source Code.

OpenVision Technologies, Inc. has donated this Kerberos Administration system to MIT for inclusion in the standard Kerberos 5 distribution. This donation underscores our commitment to continuing Kerberos technology development and our gratitude for the valuable work which has been performed by MIT and the Kerberos community.

=====

Portions contributed by Matt Crawford "crawdada@fnal.gov" were work performed at Fermi National Accelerator Laboratory, which is operated by Universities Research Association, Inc., under contract DE-AC02-76CHO3000 with the U.S. Department of Energy.

=====

Portions of "src/lib/crypto" have the following copyright:

Copyright (C) 1998 by the FundsXpress, INC.

All rights reserved.

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of FundsXpress. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. FundsXpress makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

=====

The implementation of the AES encryption algorithm in "src/lib/crypto/builtin/aes" has the following copyright:

Copyright (C) 1998-2013, Brian Gladman, Worcester, UK. All rights reserved.

The redistribution and use of this software (with or without changes) is allowed without the payment of fees or royalties provided that:

source code distributions include the above copyright notice,  
this list of conditions and the following disclaimer;

binary distributions include the above copyright notice, this  
list of conditions and the following disclaimer in their  
documentation.

This software is provided 'as is' with no explicit or implied  
warranties in respect of its operation, including, but not limited  
to, correctness and fitness for purpose.

=====

Portions contributed by Red Hat, including the pre-authentication  
plug-in framework and the NSS crypto implementation, contain the  
following copyright:

Copyright (C) 2006 Red Hat, Inc.  
Portions copyright (C) 2006 Massachusetts Institute of Technology  
All Rights Reserved.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions  
are met:

- \* Redistributions of source code must retain the above copyright  
notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above  
copyright notice, this list of conditions and the following  
disclaimer in the documentation and/or other materials provided  
with the distribution.
- \* Neither the name of Red Hat, Inc., nor the names of its  
contributors may be used to endorse or promote products derived  
from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS  
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT  
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS  
FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE  
COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,  
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES  
(INCLUDING, BUT NOT LIMITED  
TO, PROCUREMENT OF SUBSTITUTE GOODS OR  
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,  
STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)



ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

The bundled verto source code is subject to the following license:

Copyright 2011 Red Hat, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

The MS-KKDCP client implementation has the following copyright:

Copyright 2013,2014 Red Hat, Inc.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS  
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT  
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS  
FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE  
COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,  
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES  
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR  
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,  
STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE  
OR OTHERWISE)  
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED  
OF THE POSSIBILITY OF SUCH DAMAGE.

=====

The implementations of GSSAPI mechglue in GSSAPI-SPNEGO in  
"src/lib/gssapi", including the following files:

- lib/gssapi/generic/gssapi\_err\_generic.et
- lib/gssapi/mechglue/g\_accept\_sec\_context.c
- lib/gssapi/mechglue/g\_acquire\_cred.c
- lib/gssapi/mechglue/g\_canon\_name.c
- lib/gssapi/mechglue/g\_compare\_name.c
- lib/gssapi/mechglue/g\_context\_time.c
- lib/gssapi/mechglue/g\_delete\_sec\_context.c
- lib/gssapi/mechglue/g\_dsp\_name.c
- lib/gssapi/mechglue/g\_dsp\_status.c
- lib/gssapi/mechglue/g\_dup\_name.c
- lib/gssapi/mechglue/g\_exp\_sec\_context.c
- lib/gssapi/mechglue/g\_export\_name.c
- lib/gssapi/mechglue/g\_glue.c
- lib/gssapi/mechglue/g\_imp\_name.c
- lib/gssapi/mechglue/g\_imp\_sec\_context.c
- lib/gssapi/mechglue/g\_init\_sec\_context.c
- lib/gssapi/mechglue/g\_initialize.c
- lib/gssapi/mechglue/g\_inquire\_context.c
- lib/gssapi/mechglue/g\_inquire\_cred.c
- lib/gssapi/mechglue/g\_inquire\_names.c
- lib/gssapi/mechglue/g\_process\_context.c
- lib/gssapi/mechglue/g\_rel\_buffer.c
- lib/gssapi/mechglue/g\_rel\_cred.c
- lib/gssapi/mechglue/g\_rel\_name.c
- lib/gssapi/mechglue/g\_rel\_oid\_set.c
- lib/gssapi/mechglue/g\_seal.c
- lib/gssapi/mechglue/g\_sign.c
- lib/gssapi/mechglue/g\_store\_cred.c

lib/gssapi/mechglue/g\_unseal.c  
lib/gssapi/mechglue/g\_userok.c  
lib/gssapi/mechglue/g\_utils.c  
lib/gssapi/mechglue/g\_verify.c  
lib/gssapi/mechglue/gssd\_pname\_to\_uid.c  
lib/gssapi/mechglue/mglueP.h  
lib/gssapi/mechglue/oid\_ops.c  
lib/gssapi/spnego/gssapiP\_spnego.h  
lib/gssapi/spnego/spnego\_mech.c

and the initial implementation of incremental propagation, including the following new or changed files:

include/iprof\_hdr.h  
kadmin/server/iprof\_svc.c  
lib/kdb/iprof.x  
lib/kdb/kdb\_convert.c  
lib/kdb/kdb\_log.c  
lib/kdb/kdb\_log.h  
lib/krb5/error\_tables/kdb5\_err.et  
kprop/kpropd\_rpc.c  
kprop/kproplog.c

are subject to the following license:

Copyright (C) 2004 Sun Microsystems, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

Kerberos V5 includes documentation and software developed at the University of California at Berkeley, which includes this copyright notice:

Copyright (C) 1983 Regents of the University of California.  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

Portions contributed by Novell, Inc., including the LDAP database backend, are subject to the following license:

Copyright (C) 2004-2005, Novell, Inc.  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* The copyright holder's name is not used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

Portions funded by Sandia National Laboratory and developed by the University of Michigan's Center for Information Technology Integration, including the PKINIT implementation, are subject to the following license:

COPYRIGHT (C) 2006-2007  
THE REGENTS OF THE UNIVERSITY OF MICHIGAN  
ALL RIGHTS RESERVED

Permission is granted to use, copy, create derivative works and redistribute this software and such derivative works for any purpose, so long as the name of The University of Michigan is not used in any advertising or publicity pertaining to the use of distribution of this software without specific, written prior authorization. If the above copyright notice or any other identification of the University of Michigan is included in any

copy of any portion of this software, then the disclaimer below must also be included.

THIS SOFTWARE IS PROVIDED AS IS, WITHOUT REPRESENTATION FROM THE UNIVERSITY OF MICHIGAN AS TO ITS FITNESS FOR ANY PURPOSE, AND WITHOUT WARRANTY BY THE UNIVERSITY OF MICHIGAN OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

THE REGENTS OF THE UNIVERSITY OF MICHIGAN SHALL NOT BE LIABLE FOR ANY DAMAGES, INCLUDING SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WITH RESPECT TO ANY CLAIM ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE SOFTWARE, EVEN IF IT HAS BEEN OR IS HEREAFTER ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

=====

The pkcs11.h file included in the PKINIT code has the following license:

Copyright 2006 g10 Code GmbH  
Copyright 2006 Andreas Jellinghaus

This file is free software; as a special exception the author gives unlimited permission to copy and/or distribute it, with or without modifications, as long as this notice is preserved.

This file is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY, to the extent permitted by law; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

=====

Portions  
contributed by Apple Inc. are subject to the following  
license:

Copyright 2004-2008 Apple Inc. All Rights Reserved.

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright

notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Apple Inc. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Apple Inc. makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

=====

The implementations of UTF-8 string handling in src/util/support and src/lib/krb5/unicode are subject to the following copyright and permission notice:

The OpenLDAP Public License  
Version 2.8, 17 August 2003

Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that the following conditions are met:

1. Redistributions in source form must retain copyright statements and notices,
2. Redistributions in binary form must reproduce applicable copyright statements and notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution, and
3. Redistributions must contain a verbatim copy of this document.

The OpenLDAP Foundation may revise this license from time to time. Each revision is distinguished by a version number. You may use this Software under terms of this license revision or under the terms of any subsequent revision of the license.

THIS SOFTWARE IS PROVIDED BY THE OPENLDAP FOUNDATION AND ITS CONTRIBUTORS "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OPENLDAP FOUNDATION, ITS CONTRIBUTORS, OR THE AUTHOR(S) OR OWNER(S) OF THE SOFTWARE BE

LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The names of the authors and copyright holders must not be used in advertising or otherwise to promote the sale, use or other dealing in this Software without specific, written prior permission. Title to copyright in this Software shall at all times remain with copyright holders.

OpenLDAP is a registered trademark of the OpenLDAP Foundation.

Copyright 1999-2003 The OpenLDAP Foundation, Redwood City, California, USA. All Rights Reserved. Permission to copy and distribute verbatim copies of this document is granted.

=====

Marked test programs in src/lib/krb5/krb  
have the following copyright:

Copyright (C) 2006 Kungliga Tekniska Hgskola  
(Royal Institute of Technology, Stockholm, Sweden).  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of KTH nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.



THIS SOFTWARE IS PROVIDED BY KTH AND ITS CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL KTH OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

The KCM Mach RPC definition file used on macOS has the following copyright:

Copyright (C) 2009 Kungliga Tekniska Hgskola  
(Royal Institute of Technology, Stockholm, Sweden).  
All rights reserved.

Portions Copyright (C) 2009 Apple Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

Portions of the RPC implementation in src/lib/rpc and  
src/include/gssrpc have the following copyright and permission notice:

Copyright (C) 2010, Oracle America, Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions  
are met:

1. Redistributions of source code must retain the above  
copyright  
notice, this list of conditions and the following  
disclaimer.
2. Redistributions in binary form must reproduce the above  
copyright notice, this list of conditions and the following  
disclaimer in the documentation and/or other materials provided  
with the distribution.
3. Neither the name of the "Oracle America, Inc." nor the names  
of its contributors may be used to endorse or promote products  
derived from this software without specific prior written  
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS  
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT  
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS  
FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE  
COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,  
INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES  
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR  
SERVICES;  
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,  
STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)  
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED  
OF THE POSSIBILITY OF SUCH DAMAGE.

=====  
Copyright (C) 2006,2007,2009 NTT (Nippon Telegraph and Telephone Corporation). All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer as the first lines of this file unmodified.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY NTT "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL NTT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====  
Copyright 2000 by Carnegie Mellon University

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Carnegie Mellon University not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

CARNEGIE MELLON UNIVERSITY DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL CARNEGIE MELLON UNIVERSITY BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

=====

Copyright

(C) 2002 Naval Research Laboratory (NRL/CCS)

Permission to use, copy, modify and distribute this software and its documentation is hereby granted, provided that both the copyright notice and this permission notice appear in all copies of the software, derivative works or modified versions, and any portions thereof.

NRL ALLOWS FREE USE OF THIS SOFTWARE IN ITS "AS IS" CONDITION AND DISCLAIMS ANY LIABILITY OF ANY KIND FOR ANY DAMAGES WHATSOEVER RESULTING FROM THE USE OF THIS SOFTWARE.

=====

Copyright (C) 2022 United States Government as represented by the Secretary of the Navy. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES

(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

Copyright (C) 1991, 1992, 1994 by Cygnus Support.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Cygnus Support makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

=====

Copyright (C) 2006 Secure Endpoints Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Portions of the implementation of the Fortuna-like PRNG are subject to the following notice:

Copyright (C) 2005  
Marko Kreen  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 1994 by the University of Southern California

EXPORT OF THIS SOFTWARE from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to copy, modify, and distribute this software and its documentation in source and binary forms is hereby granted, provided that any documentation or other materials related to such distribution or use acknowledge that the software was developed by the University of Southern California.

DISCLAIMER OF WARRANTY. THIS SOFTWARE IS PROVIDED "AS IS". The University of Southern California MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. By way of example, but not limitation, the University of Southern California MAKES NO REPRESENTATIONS OR WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. The University of Southern California shall not be held liable for any liability nor for any direct, indirect, or consequential damages with respect to any claim by the user or distributor of the ksu software.

=====

Copyright (C) 1995  
The President and Fellows of Harvard University

This code is derived from software contributed to Harvard by Jeremy Rassen.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:  
  
This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

Copyright (C) 2008 by the Massachusetts Institute of Technology.

Copyright 1995 by Richard P. Basch. All Rights Reserved.

Copyright 1995 by Lehman Brothers, Inc. All Rights Reserved.

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Richard P. Basch, Lehman Brothers and M.I.T. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Richard P. Basch, Lehman Brothers and M.I.T. make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

=====

The following notice applies to "src/lib/krb5/krb/strptime.c" and "src/include/k5-queue.h".

Copyright (C) 1997, 1998 The NetBSD Foundation, Inc.

All rights reserved.

This code was contributed to The NetBSD Foundation by Klaus Klein.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.



2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

This product includes software developed by the NetBSD Foundation, Inc. and its contributors.

4. Neither the name of The NetBSD Foundation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

The  
following notice applies to Unicode library files in  
"src/lib/krb5/unicode":

Copyright 1997, 1998, 1999 Computing Research Labs,  
New Mexico State University

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN

NO EVENT SHALL THE COMPUTING RESEARCH LAB OR NEW MEXICO STATE UNIVERSITY BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

The following notice applies to "src/util/support/strncpy.c":

Copyright (C) 1998 Todd C. Miller "Todd.Miller@courtesan.com"

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

=====

The following notice applies to "src/util/profile/argv\_parse.c" and "src/util/profile/argv\_parse.h":

Copyright 1999 by Theodore Ts'o.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies. THE SOFTWARE IS PROVIDED "AS IS" AND THEODORE TS'O (THE AUTHOR) DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT,

INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER  
RESULTING FROM LOSS  
OF USE, DATA OR PROFITS, WHETHER IN AN ACTION  
OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR  
IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. (Isn't  
it sick that the U.S. culture of lawsuit-happy lawyers requires  
this kind of disclaimer?)

=====

The following notice applies to SWIG-generated code in  
"src/util/profile/profile\_tcl.c":

Copyright (C) 1999-2000, The University of Chicago

This file may be freely redistributed without license or fee  
provided this copyright message remains intact.

=====

The following notice applies to portions of "src/lib/rpc" and  
"src/include/gssrpc":

Copyright (C) 2000 The Regents of the University of Michigan. All  
rights reserved.

Copyright (C) 2000 Dug Song "dugsong@UMICH.EDU". All rights  
reserved, all wrongs reversed.

Redistribution and use in source  
and binary forms, with or without  
modification, are permitted provided that the following conditions  
are met:

1. Redistributions of source code must retain the above  
copyright notice, this list of conditions and the following  
disclaimer.
2. Redistributions in binary form must reproduce the above  
copyright notice, this list of conditions and the following  
disclaimer in the documentation and/or other materials provided  
with the distribution.
3. Neither the name of the University nor the names of its  
contributors may be used to endorse or promote products derived  
from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED

WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

Implementations of the MD4 algorithm are subject to the following notice:

Copyright (C) 1990, RSA Data Security, Inc. All rights reserved.

License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD4 Message Digest Algorithm" in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD4 Message Digest Algorithm" in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

=====

Implementations of the MD5 algorithm are subject to the following notice:

Copyright (C) 1990, RSA Data Security, Inc. All rights reserved.

License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message- Digest

Algorithm" in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

=====

The following notice applies to  
"src/lib/crypto/crypto\_tests/t\_md5driver.c":

Copyright (C) 1990-2, RSA Data Security, Inc. Created 1990. All rights reserved.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

=====

Portions of "src/lib/krb5" are subject to the following notice:

Copyright (C) 1994 CyberSAFE Corporation.  
Copyright 1990,1991,2007,2008 by the Massachusetts  
Institute of Technology.  
All Rights Reserved.

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of M.I.T. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Furthermore if you modify this software you must label your software as modified software and not distribute it in such a fashion that it might be confused with the original M.I.T. software. Neither M.I.T., the Open Computing Security Group, nor CyberSAFE Corporation make any representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

=====

Portions contributed by PADL Software are subject to the following license:

Copyright (c) 2011, PADL Software Pty Ltd. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions

of source code must retain the above

copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above

copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of PADL Software nor the names of its

contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY PADL SOFTWARE AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL PADL SOFTWARE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT

OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

The bundled libev source code is subject to the following license:

All files in libev are Copyright (C)2007,2008,2009 Marc Alexander Lehmann.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Alternatively, the contents of this package may be used under the terms of the GNU General Public License ("GPL") version 2 or any

later version, in which case the provisions of the GPL are applicable instead of the above. If you wish to allow the use of your version of this package only under the terms of the GPL and not to allow others to use your version of this file under the BSD license, indicate your decision by deleting the provisions above

and replace them with the notice and other provisions required by the GPL in this and the other files of this package. If you do not delete the provisions above, a recipient may use your version of this file under either the BSD or the GPL.

=====

Files copied from the Intel AESNI Sample Library are subject to the following license:

Copyright (C) 2010, Intel Corporation All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Intel Corporation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

The following notice applies to



"src/ccapi/common/win/OldCC/autolock.hxx":

Copyright (C) 1998 by Danilo Almeida. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer

in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

The following notice applies to portions of

"src/plugins/preauth/spake/edwards25519.c"

and

"src/plugins/preauth/spake/edwards25519\_tables.h":

The MIT License (MIT)

Copyright (c) 2015-2016 the fiat-crypto authors (see the AUTHORS file).

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to

the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

The following notice applies to portions of  
"src/plugins/preauth/spake/edwards25519.c":

Copyright (c) 2015-2016, Google Inc.

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

.. \_mitK5license:

MIT Kerberos License information

=====

.. toctree::  
:hidden:

copyright.rst

.. include:: notice.rst

# 1.91 importlib-metadata 8.0.0

## 1.91.1 Available under license :

UNRECOGNIZED LICENSE; MD5 sum: 67a0449e1745cb3a0d5ef058e15bb91a

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the

editorial  
revisions, annotations, elaborations, or other modifications  
represent, as a whole, an original work of authorship. For the purposes  
of this License, Derivative Works shall not include works that remain  
separable from, or merely link (or bind by name) to the interfaces of,  
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including  
the original version of the Work and any modifications or additions  
to that Work or Derivative Works thereof, that is intentionally  
submitted to Licensor for inclusion in the Work by the copyright owner  
or by an individual or Legal Entity authorized to submit on behalf of  
the copyright owner. For the purposes of this definition, "submitted"  
means any form of electronic, verbal, or written communication sent  
to the Licensor or its representatives, including but not limited to  
communication on electronic mailing lists, source code control systems,  
and issue tracking systems that are managed by, or on behalf of, the  
Licensor for the purpose of discussing and improving the Work, but  
excluding communication that is conspicuously marked or otherwise  
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity  
on behalf of whom a Contribution has been received by Licensor and  
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of  
this License, each Contributor hereby grants to You a perpetual,  
worldwide, non-exclusive, no-charge, royalty-free, irrevocable  
copyright license to reproduce, prepare Derivative Works of,  
publicly display, publicly perform, sublicense, and distribute the  
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of  
this  
License, each Contributor hereby grants to You a perpetual,  
worldwide, non-exclusive, no-charge, royalty-free, irrevocable  
(except as stated in this section) patent license to make, have made,  
use, offer to sell, sell, import, and otherwise transfer the Work,  
where such license applies only to those patent claims licensable  
by such Contributor that are necessarily infringed by their  
Contribution(s) alone or by combination of their Contribution(s)  
with the Work to which such Contribution(s) was submitted. If You  
institute patent litigation against any entity (including a  
cross-claim or counterclaim in a lawsuit) alleging that the Work  
or a Contribution incorporated within the Work constitutes direct  
or contributory patent infringement, then any patent licenses  
granted to You under this License for that Work shall terminate  
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute

copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of

this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## 1.92 libmaxminddb 1.9.1-r0

### 1.92.1 Available under license :

Copyright 2013-2023 MaxMind, Inc.

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

GNU LESSER GENERAL PUBLIC LICENSE  
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

## 0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

## 1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

## 2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:



a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

### 3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

### 4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

## 5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

## 6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions

of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,

including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of

this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/> or send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

# 1.93 iperf 2.2.0-r0

## 1.93.1 Notifications :

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)

This product includes cryptographic software written by Eric Young ([ey@cryptsoft.com](mailto:ey@cryptsoft.com)).

This product includes software written by Tim Hudson ([tjh@cryptsoft.com](mailto:tjh@cryptsoft.com)).

## 1.93.2 Available under license :

"iperf, Copyright (c) 2014-2020, The Regents of the University of California, through Lawrence Berkeley National Laboratory (subject to receipt of any required approvals from the U.S. Dept. of Energy). All rights reserved."

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

(1) Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

(2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

(3) Neither the name of the University of California, Lawrence Berkeley National Laboratory, U.S. Dept. of Energy nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY

THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND

ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You are under no obligation whatsoever to provide any bug fixes, patches, or upgrades to the features, functionality or performance of the source code ("Enhancements") to anyone; however, if you choose to make your Enhancements available either publicly, or directly to Lawrence

Berkeley National

Laboratory, without imposing a separate written license agreement for such Enhancements, then you hereby grant the following license: a non-exclusive, royalty-free perpetual license to install, use, modify, prepare derivative works, incorporate into other computer software, distribute, and sublicense such enhancements or derivative works thereof, in binary and source code form.



=====

This software contains source code (src/cjson.{c,h}) that is:

Copyright (c) 2009-2017 Dave Gamble and cJSON contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

This software contains source code (src/net.{c,h}) that is:

This software was developed as part of a project at MIT.

Copyright (c) 2005-2007 Russ Cox,  
Massachusetts Institute of Technology

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,

EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

===

Contains parts of an earlier library that has:

/\*

\* The authors of this software are Rob Pike, Sape Mullender, and Russ Cox

\* Copyright (c) 2003 by Lucent Technologies.

\* Permission to use, copy, modify, and distribute this software for any

\* purpose without fee is hereby granted, provided that this entire notice

\* is included in all copies of any software which is or includes a copy

\* or modification of this software and in all copies of the supporting

\* documentation for such software.

\* THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED

\* WARRANTY. IN PARTICULAR, NEITHER THE AUTHORS NOR LUCENT TECHNOLOGIES MAKE ANY

\* REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY

\* OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

\*/

=====

This software contains source code (src/net.c) that is:

/\*

\* Copyright (c) 2001 Eric Jackson <ericj@monkey.org>

\*

\* Redistribution and use in source and binary forms, with or without

\* modification, are permitted provided that the following conditions

\* are met:

\*

\* 1. Redistributions

of source code must retain the above copyright

\* notice, this list of conditions and the following disclaimer.

\* 2. Redistributions in binary form must reproduce the above copyright

\* notice, this list of conditions and the following disclaimer in the

\* documentation and/or other materials provided with the distribution.

\* 3. The name of the author may not be used to endorse or promote products

\* derived from this software without specific prior written permission.

\*

\* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR

\* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES  
 \* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.  
 \* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,  
 \* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT  
 \* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,  
 \* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED  
 AND ON ANY  
 \* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT  
 \* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF  
 \* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.  
 \*/

=====

This software contains source code (src/queue.h) that is:

```
/*
 * Copyright (c) 1991, 1993
 *   The Regents of the University of California.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *   notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *   notice, this list of conditions and the following disclaimer in the
 *   documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the University nor the names of
its contributors
 *   may be used to endorse or promote products derived from this software
 *   without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 *   @(#)queue.h   8.5 (Berkeley) 8/20/94
 */
```

=====

This software contains source code (src/units.{c,h}) that is:

```
/*-----  
* Copyright (c) 1999,2000,2001,2002,2003  
* The Board of Trustees of the University of Illinois  
* All Rights Reserved.  
*-----  
* Permission is hereby granted, free of charge, to any person  
* obtaining a copy of this software (Iperf) and associated  
* documentation files (the "Software"), to deal in the Software  
* without restriction, including without limitation the  
* rights to use, copy, modify, merge, publish, distribute,  
* sublicense, and/or sell copies of the Software, and to permit  
* persons to whom the Software is furnished to do  
* so, subject to the following conditions:  
*  
*  
* Redistributions of source code must retain the above  
* copyright notice, this list of conditions and  
* the following disclaimers.  
*  
*  
* Redistributions  
in binary form must reproduce the above  
* copyright notice, this list of conditions and the following  
* disclaimers in the documentation and/or other materials  
* provided with the distribution.  
*  
*  
* Neither the names of the University of Illinois, NCSA,  
* nor the names of its contributors may be used to endorse  
* or promote products derived from this Software without  
* specific prior written permission.  
*  
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,  
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES  
* OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND  
* NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT  
* HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,  
* WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,  
* ARISING FROM, OUT OF OR IN CONNECTION WITH THE  
* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.  
*  
*-----  
* National Laboratory for Applied Network Research  
* National Center for Supercomputing Applications  
* University of Illinois at Urbana-Champaign
```

\* <http://www.ncsa.uiuc.edu>

\*

\*

\* `stdio.c`

\* by Mark Gates <[mgates@nlanr.net](mailto:mgates@nlanr.net)>

\* and Ajay Tirumalla <[tirumala@ncsa.uiuc.edu](mailto:tirumala@ncsa.uiuc.edu)>

\* -----

\* input and output numbers, converting with kilo, mega, giga

\* ----- \*/

=====

This software contains source code (`src/portable_endian.h`) that is:

// "License": Public Domain

// I, Mathias Panzenbck, place this file hereby into the public domain. Use it at your own risk for whatever you like.

=====

Copyright (c) 2014, ESnet

All rights reserved.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions are met:

\* Redistributions of source code must retain the above copyright notice, this  
list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice,  
this list of conditions and the following disclaimer in the documentation  
and/or other materials provided with the distribution.

\* Neither the name of the {organization} nor the names of its  
contributors may be used to endorse or promote products derived from  
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"  
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE  
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER  
OR CONTRIBUTORS BE LIABLE  
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL  
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR  
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER  
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,  
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE  
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# 1.94 nghttp2 1.65.0-r0

## 1.94.1 Available under license :

The MIT License

Copyright (c) 2012, 2014, 2015, 2016 Tatsuhiro Tsujikawa

Copyright (c) 2012, 2014, 2015, 2016 nghttp2 contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

See COPYING

This software is licensed under the MIT License.

Copyright Fedor Indutny, 2018.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,

DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,

ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

If not otherwise noted, the extensions in this package are licensed under the following license.

Copyright (c) 2010 by the contributors (see AUTHORS file).

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

\* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,

EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.95 apk-tools 2.14.9-r3

### 1.95.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification



follow.

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most

ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works.

But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it

with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all

the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as

a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation,

write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

## NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE

THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

## How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and an idea of what it does.>

Copyright (C)< yyyy> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon >, 1 April 1989 Ty Coon, President of Vice This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

## 1.96 netcat-openbsd 1.229.1-r0

### 1.96.1 Available under license :

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (c) 2006 Damien Miller. All rights reserved.

\* Copyright (c) 2005 Anil Madhavapeddy. All rights reserved.  
 \* Copyright (c) 1995,1999 Theo de Raadt. All rights reserved.  
 \* All rights reserved.  
 \*  
 \* Redistribution and use in source and binary forms, with or without  
 \* modification, are permitted provided that the following conditions  
 \* are met:  
 \* 1. Redistributions of source code must retain the above copyright  
 \* notice, this list of conditions and the following disclaimer.  
 \* 2. Redistributions in binary form must reproduce the above copyright  
 \* notice, this list of conditions and the following disclaimer in the  
 \* documentation and/or other materials provided with the distribution.  
 \*  
 \* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR  
 \* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES  
 \* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
 ARE DISCLAIMED.  
 \* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,  
 \* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT  
 \* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,  
 \* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY  
 \* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT  
 \* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF  
 \* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.  
 \*/

Found in path(s):

\* /netcat-openbsd-1-229-orig-tar/netcat-openbsd-1.229/atomicio.c  
 No license file was found, but licenses were detected in source scan.

/\*  
 \* Copyright (c) 2006 Damien Miller. All rights reserved.  
 \* Copyright (c) 1995,1999 Theo de Raadt. All rights reserved.  
 \* All rights reserved.  
 \*  
 \* Redistribution and use in source and binary forms, with or without  
 \* modification, are permitted provided that the following conditions  
 \* are met:  
 \* 1. Redistributions of source code must retain the above copyright  
 \* notice, this list of conditions and the following disclaimer.  
 \* 2. Redistributions in binary form must reproduce the above copyright  
 \* notice, this list of conditions and the following disclaimer in the  
 \* documentation and/or other materials provided with the distribution.  
 \*  
 \* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR  
 \* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES  
 \* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.  
 \* IN NO EVENT SHALL THE AUTHOR BE LIABLE

FOR ANY DIRECT, INDIRECT,

\* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT  
\* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,  
\* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY  
\* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT  
\* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF  
\* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.  
\*/

Found in path(s):

\* /netcat-openbsd-1-229-orig-tar/netcat-openbsd-1.229/atomicio.h

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (c) 2001 Eric Jackson <ericj@monkey.org>

\* Copyright (c) 2015 Bob Beck. All rights reserved.

\*

\* Redistribution and use in source and binary forms, with or without

\* modification, are permitted provided that the following conditions

\* are met:

\*

\* 1. Redistributions of source code must retain the above copyright

\* notice, this list of conditions and the following disclaimer.

\* 2. Redistributions in binary form must reproduce the above copyright

\* notice, this list of conditions and the following disclaimer in the

\* documentation and/or other materials provided with the distribution.

\* 3. The name of the author may not be used to endorse or promote products

\* derived from this software without specific prior written permission.

\*

\* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR

\* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

\*

OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

\* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,

\* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT

\* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

\* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

\* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

\* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF

\* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

\*/

Found in path(s):

\* /netcat-openbsd-1-229-orig-tar/netcat-openbsd-1.229/netcat.c

No license file was found, but licenses were detected in source scan.

/\*

\* Copyright (c) 1999 Niklas Hallqvist. All rights reserved.



```

* Copyright (c) 2004, 2005 Damien Miller. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```

Found in path(s):

```
* /netcat-openbsd-1-229-orig-tar/netcat-openbsd-1.229/socks.c
```

No license file was found, but licenses were detected in source scan.

```

.\" All rights reserved.
.\" Redistribution and use in source and binary forms, with or without
.\" modification, are permitted provided that the following conditions
.\" are met:
.\" 1. Redistributions of source code must retain the above copyright
.\" notice, this list of conditions and the following disclaimer.
.\" 2. Redistributions in binary form must reproduce the above copyright
.\" notice, this list of conditions and the following disclaimer in the
.\" documentation and/or other materials provided with the distribution.
.\" 3. The name of the author may not be used to endorse or promote products

```

Found in path(s):

```
* /netcat-openbsd-1-229-orig-tar/netcat-openbsd-1.229/nc.1
```

## 1.97 fdisk 2.41.0

### 1.97.1 Available under license :

|  | NR | START | END   | SECTORS | SIZE | NAME | UUID        |
|--|----|-------|-------|---------|------|------|-------------|
|  | 1  | 32    | 7679  | 7648    | 3.7M |      | 8f8378c0-01 |
|  | 2  | 7680  | 16383 | 8704    | 4.3M |      | 8f8378c0-02 |

5 7936 12799 4864 2.4M

6 12544 16127 3584 1.8M

## GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<https://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1)

assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains

that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS

### 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

#### 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The

"System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to

control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

## 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

## 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological

measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

#### 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately  
publish on each copy an appropriate copyright notice;  
keep intact all notices stating that this License and any  
non-permissive terms added in accord with section 7 apply to the code;  
keep intact all notices of the absence of any warranty; and give all  
recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

#### 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to  
"keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not

invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

## 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or



specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

## 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

## 8. Termination.

You may not propagate or modify a covered work except as expressly

provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

#### 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

#### 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that

transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights

granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

#### 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this

License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

## 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may

not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

### 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

### 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

### 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT

HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17.

Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify

it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<https://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <<https://www.gnu.org/licenses/why-not-lgpl.html>>.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next paragraph) shall be included in all copies or substantial portions of the Software.



THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, and the entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ALL OF WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF NOT ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public

License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

The complete text of the license is available in the  
../Documentation/licenses/COPYING.LGPL-2.1-or-later file.

/\*

\* Copyright (c) 1989 The Regents of the University of California.

\* All rights reserved.

\*

\* Redistribution and use in source and binary forms, with or without

\* modification, are permitted provided that the following conditions

\* are met:

\* 1. Redistributions of source code must retain the above copyright

\* notice, this list of conditions and the following disclaimer.

\* 2. Redistributions in binary form must reproduce the above copyright

\* notice, this list of conditions and the following disclaimer in the

\* documentation and/or other materials provided with the distribution.

\* 3. All advertising materials mentioning features or use of this software

\* must display the following acknowledgement:

\* This product includes software developed by the University of

\* California, Berkeley and its contributors.

\* 4. Neither the name of the University nor the names of its contributors

\* may be used to endorse or promote products derived from this software

\*

without specific prior written permission.

\*

\* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND

\* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

\* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

\* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE

\* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

\* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS

\* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

\* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT

\* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY

\* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF

\* SUCH DAMAGE.

\*/

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this  
list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,  
this list of conditions and the following disclaimer in the documentation  
and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

HOWEVER

CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#### GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the

source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate

copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt

otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made

generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF



MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR

THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute
it
under certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

#### GNU LESSER GENERAL PUBLIC LICENSE Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change

free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that

any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

## GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact

all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary  
GNU General Public

License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding  
machine-readable source code, which  
must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that



uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

- b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally

distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on

the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent

license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these,

write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

##### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!  
This library is free software; you can redistribute it and/or modify it under the terms of the Modified BSD License.

The complete text of the license is available in the  
../Documentation/licenses/COPYING.BSD-3-Clause file.  
BSD 2-Clause License

Copyright (c) 2023, Thorsten Kukuk

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation

and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;

LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

The complete text of the license is available in the  
../Documentation/licenses/COPYING.LGPL-2.1-or-later  
/\*

\* No copyright is claimed. This code is in the public domain; do with  
\* it what you wish.  
\*/

```
#ifndef UTIL_LINUX_PT_BSD_H
#define UTIL_LINUX_PT_BSD_H
```

```
#define BSD_MAXPARTITIONS 16
#define BSD_FS_UNUSED 0
```

```
#ifndef BSD_DISKMAGIC
# define BSD_DISKMAGIC ((uint32_t) 0x82564557)
#endif
```

```
#define BSD_LINUX_BOOTDIR "/usr/ucb/mdec"
```

```
#if defined (__alpha__) || defined (__powerpc__) || \
    defined (__ia64__) || defined (__hppa__)
# define BSD_LABELSECTOR 0
# define BSD_LABELOFFSET 64
#else
# define BSD_LABELSECTOR 1
# define BSD_LABELOFFSET 0
#endif
```

```
#define BSD_BBSIZE      8192 /* size of boot area, with label */
#define BSD_SBSIZE      8192 /* max size of fs superbblock */
```

```

struct bsd_disklabel {
    uint32_t d_magic; /* the magic number */
    int16_t d_type; /* drive type */
    int16_t d_subtype; /* controller/d_type specific */
    char d_typename[16]; /* type name, e.g. "eagle" */
    char d_packname[16]; /* pack identifier */

    /* disk geometry: */
    uint32_t d_sectsize; /*
# of bytes per sector */
    uint32_t d_nsectors; /* # of data sectors per track */
    uint32_t d_ntracks; /* # of tracks per cylinder */
    uint32_t d_ncylinders; /* # of data cylinders per unit */
    uint32_t d_secp cyl; /* # of data sectors per cylinder */
    uint32_t d_secperunit; /* # of data sectors per unit */

    /*
* Spares (bad sector replacements) below
* are not counted in d_nsectors or d_secp cyl.
* Spare sectors are assumed to be physical sectors
* which occupy space at the end of each track and/or cylinder.
*/
    uint16_t d_sparesp track; /* # of spare sectors per track */
    uint16_t d_spaersp cyl; /* # of spare sectors per cylinder */

    /*
* Alternate cylinders include maintenance, replacement,
* configuration description areas, etc.
*/
    uint32_t d_acylinders; /* # of alt. cylinders per unit */

    /* hardware characteristics: */
    /*
* d_interleave, d_trackskew and d_cylskew describe perturbations
* in the media format used to compensate for a slow
controller.
* Interleave is physical sector interleave, set up by the formatter
* or controller when formatting. When interleaving is in use,
* logically adjacent sectors are not physically contiguous,
* but instead are separated by some number of sectors.
* It is specified as the ratio of physical sectors traversed
* per logical sector. Thus an interleave of 1:1 implies contiguous
* layout, while 2:1 implies that logical sector 0 is separated
* by one sector from logical sector 1.
* d_trackskew is the offset of sector 0 on track N
* relative to sector 0 on track N-1 on the same cylinder.
* Finally, d_cylskew is the offset of sector 0 on cylinder N

```

```

* relative to sector 0 on cylinder N-1.
*/
uint16_t d_rpm; /* rotational speed */
uint16_t d_interleave; /* hardware sector interleave */
uint16_t d_trackskew; /* sector 0 skew, per track */
uint16_t d_cylskew; /* sector 0 skew, per cylinder */
uint32_t d_headswitch; /* head switch time, usec */
uint32_t d_trkseek; /*
track-to-track seek, usec */
uint32_t d_flags; /* generic flags */
uint32_t d_drivedata[5]; /* drive-type specific information */
uint32_t d_spare[5]; /* reserved for future use */
uint32_t d_magic2; /* the magic number (again) */
uint16_t d_checksum; /* xor of data incl. partitions */

/* filesystem and partition information: */
uint16_t d_npartitions; /* number of partitions in following */
uint32_t d_bbsize; /* size of boot area at sn0, bytes */
uint32_t d_sbsize; /* max size of fs superblock, bytes */

struct bsd_partition { /* the partition table */
    uint32_t p_size; /* number of sectors in partition */
    uint32_t p_offset; /* starting sector */
    uint32_t p_fsize; /* filesystem basic fragment size */
    uint8_t p_fstype; /* filesystem type, see below */
    uint8_t p_frag; /* filesystem fragments per block */
    uint16_t p_cpgr; /* filesystem cylinders per group */
} __attribute__((packed))
d_partitions[BSD_MAXPARTITIONS]; /* actually may be more */
} __attribute__((packed));

/* d_type values: */
#define BSD_DTYPE_SMD 1 /* SMD, XSMD; VAX hp/up */
#define BSD_DTYPE_MSCP 2 /* MSCP */
#define BSD_DTYPE_DEC 3 /* other DEC (rk, rl) */
#define BSD_DTYPE_SCSI 4 /* SCSI */
#define BSD_DTYPE_ESDI 5 /* ESDI interface */
#define BSD_DTYPE_ST506 6 /* ST506 etc. */
#define BSD_DTYPE_HPIB 7 /* CS/80 on HP-IB */
#define BSD_DTYPE_HPFL 8 /* HP Fiber-link */
#define BSD_DTYPE_FLOPPY 10 /* floppy */

/* d_subtype values: */
#define BSD_DSTYPE_INDOSPART 0x8 /* is inside dos partition */
#define BSD_DSTYPE_DOSPART(s) ((s) & 3) /* dos partition number */
#define BSD_DSTYPE_GEOMETRY 0x10 /* drive params in label */

```



```

/*
 * Filesystem type and version.
 * Used to interpret other filesystem-specific
 * per-partition information.
 */

#define BSD_FS_UNUSED 0 /* unused */
#define BSD_FS_SWAP 1 /* swap */
#define BSD_FS_V6 2 /* Sixth Edition */
#define BSD_FS_V7
    3 /* Seventh Edition */
#define BSD_FS_SYSV 4 /* System V */
#define BSD_FS_V71K 5 /* V7 with 1K blocks (4.1, 2.9) */
#define BSD_FS_V8 6 /* Eighth Edition, 4K blocks */
#define BSD_FS_BSDFFS 7 /* 4.2BSD fast file system */
#define BSD_FS_BSDLFS 9 /* 4.4BSD log-structured file system */
#define BSD_FS_OTHER 10 /* in use, but unknown/unsupported */
#define BSD_FS_HPFS 11 /* OS/2 high-performance file system */
#define BSD_FS_ISO9660 12 /* ISO-9660 filesystem (cdrom) */
#define BSD_FS_ISOFS BSD_FS_ISO9660
#define BSD_FS_BOOT 13 /* partition contains bootstrap */
#define BSD_FS_ADOS 14 /* AmigaDOS fast file system */
#define BSD_FS_HFS 15 /* Macintosh HFS */
#define BSD_FS_ADVFS 16 /* Digital Unix AdvFS */

/* this is annoying, but it's also the way it is :( */
#ifdef __alpha__
#define BSD_FS_EXT2 8 /* ext2 file system */
#else
#define BSD_FS_MSDOS 8 /* MS-DOS file system */
#endif

/*
 * flags shared by various drives:
 */
#define BSD_D_REMOVABLE 0x01 /*
    removable media */
#define BSD_D_ECC 0x02 /* supports ECC */
#define BSD_D_BADSECT 0x04 /* supports bad sector forw. */
#define BSD_D_RAMDISK 0x08 /* disk emulator */
#define BSD_D_CHAIN 0x10 /* can do back-back transfers */
#define BSD_D_DOSPART 0x20 /* within MSDOS partition */

#endif /* UTIL_LINUX_PT_BSD_H */

```

## 1.98 libtasn 4.20.0-r0

## 1.98.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<https://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1)

assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and

authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand

ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS

### 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

#### 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The

"System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's

System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

## 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

## 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article

11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

#### 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately

- publish on each copy an appropriate copyright notice;
- keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code;
- keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

#### 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

## 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as

long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as



part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

## 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty

or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on

those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

## 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or

modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

#### 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

#### 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever

licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights

granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

## 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this

License, through a publicly available network server or other readily accessible means,

then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

## 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey

the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

### 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

### 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

### 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS"

WITHOUT WARRANTY

OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17.

Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by

the Free Software Foundation, either version 3 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.

You should have received a copy of the GNU General Public License  
along with this program. If not, see <<https://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short  
notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>  
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate  
parts of the General Public License. Of course, your program's commands  
might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school,  
if any, to sign a "copyright disclaimer" for the program, if necessary.  
For more information on this, and how to apply and follow the GNU GPL, see  
<<https://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program  
into proprietary programs. If your program is a subroutine library, you  
may consider it more useful to permit linking proprietary applications with  
the library. If this is what you want  
to do, use the GNU Lesser General  
Public License instead of this License. But first, please read  
<<https://www.gnu.org/licenses/why-not-lgpl.html>>.

GNU LESSER GENERAL PUBLIC LICENSE  
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
<<https://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts  
as the successor of the GNU Library Public License, version 2, hence  
the version number 2.1.]



## Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but

we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL  
PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility

in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those

sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License.

Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6.

Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not



excuse you from the conditions of this License.

If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser

General Public License from time to time.

Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a

license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

##### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is

safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, see <<https://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Moe Ghoul>, 1 April 1990  
Moe Ghoul, President of Vice

That's all there is to it!  
@c The GNU Free Documentation License.  
@center Version 1.3, 3 November 2008

@c This file is intended to be included within another document,  
@c hence no sectioning command or @node.

@display  
Copyright @copyright{} 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.  
@uref{<http://fsf.org/>}

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.  
@end display

@enumerate 0

@item

## PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document @dfn{free} in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of ``copyleft'', which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

@item

## APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The ``Document'', below, refers

to any such manual or work. Any member of the public is a licensee, and is addressed as ``you''. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A ``Modified Version'' of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A ``Secondary Section'' is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall

subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The ``Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The ``Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A ``Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not ``Transparent" is called ``Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format,  $\text{LaTeX}$  input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG@. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for

output purposes only.

The ``Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, ``Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The ``publisher" means any person or entity that distributes copies of the Document to the public.

A section ``Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as ``Acknowledgements", ``Dedications", ``Endorsements", or ``History".) To ``Preserve the Title" of such a section when you modify the Document means that it remains a section ``Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

@item

## VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

@item

## COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

@item  
MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

@enumerate A

@item

Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

@item

List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

@item

State on the Title page the name of the publisher of the Modified Version, as the publisher.

@item

Preserve all the copyright notices of the Document.

@item

Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

@item

Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

@item

Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

@item

Include an unaltered copy of this License.

@item

Preserve the section Entitled ``History'', Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled ``History'' in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.



@item

Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the ``History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

@item

For any section Entitled ``Acknowledgements" or ``Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

@item

Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

@item

Delete any section Entitled ``Endorsements". Such a section may not be included in the Modified Version.

@item

Do not retitle any existing section to be Entitled ``Endorsements" or to conflict in title with any Invariant Section.

@item

Preserve any Warranty Disclaimers.

@end enumerate

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled ``Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties---for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a

passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

@item

## COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number.

Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled ``History" in the various original documents, forming one section Entitled ``History"; likewise combine any sections Entitled ``Acknowledgements", and any sections Entitled ``Dedications". You must delete all sections Entitled ``Endorsements."

@item

## COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

@item

## AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an ``aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form.

Otherwise they must appear on printed covers that bracket the whole aggregate.

@item

## TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled ``Acknowledgements", ``Dedications", or ``History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual

title.

@item

## TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

@item

## FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See [@uref{http://www.gnu.org/copyleft/}](http://www.gnu.org/copyleft/).

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License ``or any later version'' applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not

as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

@item

RELICENSING

``Massive Multiauthor Collaboration Site" (or ``MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A ``Massive Multiauthor Collaboration" (or ``MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

``CC-BY-SA" means the Creative Commons

Attribution-Share Alike 3.0

license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

``Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is ``eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

@end enumerate

@page

@heading ADDENDUM: How to use this License for your documents

To use this License

in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

@smallexample

```

@group
Copyright (C) @var{year} @var{your name}.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts. A copy of the license is included in the section entitled ``GNU
Free Documentation License".
@end group
@end smallexample

```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the ``with@dots{ }Texts."@: line with this:

```

@smallexample
@group
  with the Invariant Sections being @var{list their titles}, with
  the Front-Cover Texts being @var{list}, and with the Back-Cover Texts
  being @var{list}.
@end group
@end smallexample

```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

```

@c Local Variables:
@c ispell-local-pdict: "ispell-dict"
@c End:

```

## 1.99 musl-fts 1.2.7-r6

### 1.99.1 Available under license :

No license file was found, but licenses were detected in source scan.

/alpine-standard-3-19-0-x86-64-iso-zip/alpine-standard-3.19.0-x86\_64.iso: binary file matches

Found in path(s):  
 \* /bin/grep

# 1.100 rich 13.9.4

## 1.100.1 Available under license :

Copyright (c) 2020 Will McGugan

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.101 backports.tarfile 1.2.0

## 1.101.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.102 platformdirs 4.3.7

## 1.102.1 Available under license :

MIT License

Copyright (c) 2024 Jacob Hummer

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.  
MIT

# 1.103 cachecontrol 0.14.2

## 1.103.1 Available under license :

Copyright 2012-2021 Eric Larson

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.



# 1.104 cachecontrol 0.14.1

## 1.104.1 Available under license :

Copyright 2012-2021 Eric Larson

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

# 1.105 tre 1.2.5-r8

## 1.105.1 Available under license :

This is the license, copyright notice, and disclaimer for TRE, a regex  
matching package (library and tools) with support for approximate  
matching.

Copyright (c) 2001-2009 Ville Laurikari <vl@iki.fi>  
All rights reserved.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions  
are met:

1. Redistributions of source code must retain the above copyright  
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright  
notice, this list of conditions and the following disclaimer in the  
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER AND CONTRIBUTORS  
``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT  
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR  
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT  
HOLDER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,  
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT  
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# 1.106 python-pip 25.0.1

## 1.106.1 Available under license :

```
UNRECOGNIZED LICENSE; MD5 sum: 45874240865793abc3a77b0bbdf3dc51
UNRECOGNIZED LICENSE; MD5 sum: b0eaf0c960a0f5d744339da436682707
UNRECOGNIZED LICENSE; MD5 sum: 856bd8901cf18d4edc40b86012a95b26
UNRECOGNIZED LICENSE; MD5 sum: d2e8c3bce80da2a2e51cffd44205a7b1
UNRECOGNIZED LICENSE; MD5 sum: 89c981cb43c0c7a860b8a18449e92a38
UNRECOGNIZED LICENSE; MD5 sum: f7cacecf6ff92c07405436643ace4302
PKu5Z\&#license.dist-0.2.dist-info/METADATAUT ggux]MN@s
J*VT$$ ~nH-dKp>NXw
'h4xOAZu'OR#diT>?ECesh
%W6Q:'nl|^(?t){u-{Z0XPKa5Zhb'!license.dist-0.2.dist-info/RECORDUT ggux}r03I,HZK
m^2fXZ;wzmUnjWU]52+\Gfmy8sUP-"7DbuhoAUo9sM"T_ST
,9%=la1?O"+l[0]y:4o$6}D
HzPs_`1rPK5d#Ahr6s%license.dist-0.2.dist-info/RECORD.jwsUT DPmgux%N@EHq()04@:*`w].9?`$mZK'3'
}g7.hl[zMeH4 TQzI9 +g\v7+L? iOU{RC3;>>p_,7:N-f@]i#
;_*bl-X,sZzoI]4a%JQy+KI(:d=^Su`t5U@d"R7PK
d#A~XY@@ license.dist-0.2.dist-info/WHEELUT FDPmguxWheel-Version: 0.1
Packager: bdist_wheel
Root-Is-Purelib: true

PK
o@licensedist/__init__.pyUT PBmguxPKu5Z\&#license.dist-0.2.dist-
info/METADATAUTguxPKa5Zhb'!)license.dist-0.2.dist-info/RECORDUTguxPK5d#Ahr6s%Vlicense.dist-0.2.dist-
info/RECORD.jwsUTDPuxPK
d#A~XY@@ license.dist-0.2.dist-info/WHEELUTFDPuxPK
o@licensedist/__init__.pyUTPuxPK
SPDX-License-Identifier: MIT
SPDX-FileCopyrightText: 2021 Taneli Hukkinen
Licensed to PSF under a Contributor Agreement.
Copyright (c) 2020 Will McGugan
```

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Python/Jython

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

#### END OF TERMS AND CONDITIONS

UNRECOGNIZED LICENSE; MD5 sum: 969a9e986521a635141d6b548aea92ad

UNRECOGNIZED LICENSE; MD5 sum: c0f4347d81adc62c6cd62b1b85aa469b

BSD 3-Clause License

Copyright (c) 2013-2024, Kim Davies and contributors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2022 Seth Michael Larson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT

OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2006-2022 by the respective authors (see AUTHORS file).

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT

OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 2008-2011 INADA Naoki <songofacandy@gmail.com>

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,



WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

UNRECOGNIZED LICENSE; MD5 sum: 224a0356792d38f867e73fdd7ee8bc68  
argparse is (c) 2006-2009 Steven J. Bethard <steven.bethard@gmail.com>.

The argparse module was contributed to Python as of Python 2.7 and thus was licensed under the Python license. Same license applies to all files in the argparse package project.

For details about the Python License, please see doc/Python-License.txt.

## History

-----

Before (and including) argparse 1.1, the argparse package was licensed under Apache License v2.0.

After argparse 1.1, all project files from the argparse project were deleted due to license compatibility issues between Apache License 2.0 and GNU GPL v2.

The project repository then had a clean start with some files taken from Python 2.7.1, so definitely all files are under Python License now.

=====

## Contributors

=====

- \* Alex Grnholm
- \* Alice Bevan-McGregor
- \* Arfrever Frehtes Taifersar Arahesis
- \* Christophe Combelles
- \* Daniel Stutzbach
- \* Daniel Holth
- \* Hanno Schlichting
- \* Jannis Leidel
- \* Jason R. Coombs
- \* Jim Fulton
- \* Jonathan Lange
- \* Justin Azoff
- \* Lennart Regebro
- \* Marc Abramowitz
- \* Martin von Lwis
- \* Noufal Ibrahim
- \* Pete Hollobon
- \* Philip Jenvey
- \* Reinout van Rees
- \* Robert Myers
- \* Stefan H. Holek

- \* Tarek Ziad
- \* Toshio Kuratomi

If you think your name is missing, please add it (alpha order by first name)  
UNRECOGNIZED LICENSE; MD5 sum: b7f43a4fdd86a40df4c36e1b6afca602  
Copyright (c) 2011 Agendaless Consulting and Contributors.  
(<http://www.agendaless.com>), All Rights Reserved  
MIT License

Copyright (c) 2008-2020 Andrey Petrov and contributors (see CONTRIBUTORS.txt)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

Contributors

=====

- \* Alex Grnholm
- \* Alice Bevan-McGregor
- \* Arfrever Frehtes Taifersar Arahesis
- \* Christophe Combelles
- \* Daniel Stutzbach
- \* Daniel Holth
- \* Dirley Rodrigues
- \* Donald Stufft
- \* Grigory Petrov
- \* Hanno Schlichting
- \* Jannis Leidel
- \* Jason R. Coombs
- \* Jim Fulton
- \* Jonathan Lange
- \* Justin Azoff

- \* Lennart Regebro
- \* Marc Abramowitz
- \* Martin von Lwis
- \* Noufal Ibrahim
- \* Pete Hollobon
- \* Phillip J. Eby
- \* Philip Jenvey
- \* Philip Thiem
- \* Reinout van Rees
- \* Robert Myers
- \* Stefan H. Holek
- \* Tarek Ziad
- \* Toshio Kuratomi

If you think your name is missing, please add it (alpha order by first name)

UNRECOGNIZED LICENSE; MD5 sum: 83ef8a4c430ba194704fb45783887e46

UNRECOGNIZED LICENSE; MD5 sum: aa439617d32fe0ada48c0da904366da4

MIT

UNRECOGNIZED LICENSE; MD5 sum: 279f913b168e268d25ef6188c9e3ffd4

Copyright (c) 2008-present The pip developers (see AUTHORS.txt file)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

UNRECOGNIZED LICENSE; MD5 sum: 69a0cd56dc900d1143c236b05bb465ac

Copyright (c) 2008-2013 The pip developers (see AUTHORS.txt file)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to

permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

License for Bundle of CA Root Certificates (pip/cacert.pem)

=====

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

Copyright (c) 2007 Ian Bicking and Contributors

Copyright (c) 2009 Ian Bicking, The Open Planning Project

Copyright (c) 2011-2013 The virtualenv developers

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



UNRECOGNIZED LICENSE; MD5 sum: 4ffcc98b2323777b0f5c002e45250264  
The MIT License (MIT)

Copyright (c) 2017 Thomas Kluyver

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR

IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

UNRECOGNIZED LICENSE; MD5 sum: 185fbb8d66d86d1c6cc10d6b557cb3f

UNRECOGNIZED LICENSE; MD5 sum: 3fd6103100722c2ab40319eb942c4c6a

UNRECOGNIZED LICENSE; MD5 sum: a7e9b70ca59c69b9aa662e8ae7bbec00

UNRECOGNIZED LICENSE; MD5 sum: 504fdd18bdcfd3f7606b5a0f88addea3

UNRECOGNIZED LICENSE; MD5 sum: bbfcba4df1f200b2a9ca494c44800c0

#### A. HISTORY OF THE SOFTWARE

=====

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see <https://www.cwi.nl>) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see <https://www.cnri.reston.va.us>) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations, which became Zope Corporation. In 2001, the Python Software Foundation (PSF, see <https://www.python.org/psf/>) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation was a sponsoring member of the PSF.

All

Python releases are Open Source (see <https://opensource.org> for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

| Release        | Derived from | Year      | Owner      | GPL-compatible? (1) |
|----------------|--------------|-----------|------------|---------------------|
| 0.9.0 thru 1.2 |              | 1991-1995 | CWI        | yes                 |
| 1.3 thru 1.5.2 | 1.2          | 1995-1999 | CNRI       | yes                 |
| 1.6            | 1.5.2        | 2000      | CNRI       | no                  |
| 2.0            | 1.6          | 2000      | BeOpen.com | no                  |
| 1.6.1          | 1.6          | 2001      | CNRI       | yes (2)             |
| 2.1            | 2.0+1.6.1    | 2001      | PSF        | no                  |
| 2.0.1          | 2.0+1.6.1    | 2001      | PSF        | yes                 |
| 2.1.1          | 2.1+2.0.1    | 2001      | PSF        | yes                 |
| 2.1.2          | 2.1.1        | 2002      | PSF        | yes                 |
| 2.1.3          | 2.1.2        | 2002      | PSF        | yes                 |
| 2.2 and above  | 2.1.1        | 2001-now  |            |                     |
| PSF            |              |           |            | yes                 |

Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

## B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

=====

Python software and documentation are licensed under the Python Software Foundation License Version 2.

Starting with Python 3.8.6,

examples, recipes, and other code in the documentation are dual licensed under the PSF License Version 2 and the Zero-Clause BSD license.

Some software incorporated into Python is under different licenses. The licenses are listed with code falling under that license.

## PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

-----

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.
2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023 Python Software Foundation; All Rights Reserved" are retained in Python alone or in any derivative version prepared by Licensee.
3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.
4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.
5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.
6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

#### BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

-----

#### BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at <http://www.pythonlabs.com/logos.html> may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

#### CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

-----

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the internet using the following URL: <http://hdl.handle.net/1895.22/1013>".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of

the changes made  
to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

-----

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER

RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

#### ZERO-CLAUSE BSD LICENSE FOR CODE IN THE PYTHON DOCUMENTATION

-----

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

This package contains a modified version of ca-bundle.crt:

ca-bundle.crt -- Bundle of CA Root Certificates

This is a bundle of X.509 certificates of public Certificate Authorities (CA). These were automatically extracted from Mozilla's root certificates file (certdata.txt). This file can be found in the mozilla source tree: <https://hg.mozilla.org/mozilla-central/file/tip/security/nss/lib/ckfw/builtins/certdata.txt> It contains the certificates in PEM format and therefore can be directly used with curl / libcurl / php\_curl, or with an Apache+mod\_ssl webserver for SSL client authentication. Just configure this file as the SSLCACertificateFile.#

\*\*\*\*\* BEGIN LICENSE BLOCK \*\*\*\*\*

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain

one at <http://mozilla.org/MPL/2.0/>.

\*\*\*\*\* END LICENSE BLOCK \*\*\*\*\*

@(#) \$RCSfile: certdata.txt,v \$ \$Revision: 1.80 \$ \$Date: 2011/11/03 15:11:58 \$  
UNRECOGNIZED LICENSE; MD5 sum: ee4b86b7d59c7064abc30c2496ac78b9  
UNRECOGNIZED LICENSE; MD5 sum: f148932318e88b83787b4b3a0bdcf70a  
UNRECOGNIZED LICENSE; MD5 sum: 351b831d68a447eabcbaa43af18d7b41

This software is made available under the terms of \*either\* of the licenses  
found in LICENSE.APACHE or LICENSE.BSD. Contributions to this software is made  
under the terms of \*both\* these licenses.

UNRECOGNIZED LICENSE; MD5 sum: b8d0bf09a388e9b6640ef5c7c153d61d  
MIT License

Copyright (c) 2021 Taneli Hukkinen

Permission is hereby granted, free of charge, to any person obtaining a copy  
of this software and associated documentation files (the "Software"), to deal  
in the Software without restriction, including without limitation the rights  
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
copies of the Software, and to permit persons to whom the Software is  
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all  
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE  
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,  
OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE  
SOFTWARE.

Copyright (c) 2018, Tzu-ping Chung <uranusjr@gmail.com>

Permission to use, copy, modify, and distribute this software for any  
purpose with or without fee is hereby granted, provided that the above  
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES  
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF  
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR  
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES  
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN  
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF  
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

UNRECOGNIZED LICENSE; MD5 sum: c0c0185b7475d96d2c346c55c3e53ab7



Pylons Project Contributor Agreement

=====

The submitter agrees by adding his or her name within the section below named "Contributors" and submitting the resulting modified document to the canonical shared repository location for this software project (whether directly, as a user with "direct commit access", or via a "pull request"), he or she is signing a contract electronically. The submitter becomes a Contributor after a) he or she signs this document by adding their name beneath the "Contributors" section below, and b) the resulting document is accepted into the canonical version control repository.

Treatment of Account

-----

Contributor will not allow anyone other than the Contributor to use his or her username or source repository login to submit code to a Pylons Project source repository. Should Contributor become aware of any such use, Contributor will immediately by notifying Agendaless Consulting. Notification must be performed by sending an email to webmaster@agendaless.com. Until such notice is received, Contributor will be presumed to have taken all actions made through Contributor's account. If the Contributor has direct commit access, Agendaless Consulting will have complete control and discretion over capabilities assigned to Contributor's account, and may disable Contributor's account for any reason at any time.

Legal Effect of Contribution

-----

Upon submitting a change or new work to a Pylons Project source Repository (a "Contribution"), you agree to assign, and hereby do assign, a one-half interest of all right, title and interest in and to copyright and other intellectual property rights with respect to your new and original portions of the Contribution to Agendaless Consulting. You and Agendaless Consulting each agree that the other shall be free to exercise any and all exclusive rights in and to the Contribution, without accounting to one another, including without limitation, the right to license the Contribution to others under the Repoze Public License. This agreement shall run with title to the Contribution. Agendaless Consulting does not convey to you any right, title or interest in or to the Program or such portions of the Contribution that were taken from the Program. Your transmission of a submission to the Pylons Project source Repository and marks of identification concerning the Contribution itself constitute your intent to contribute and your assignment of the work in accordance with the provisions of this Agreement.

## License Terms

-----

Code committed to the Pylons Project source repository (Committed Code) must be governed by the Repoze Public License (<http://repoze.org/LICENSE.txt>, aka "the RPL") or another license acceptable to Agendaless Consulting. Until Agendaless Consulting declares in writing an acceptable license other than the RPL, only the RPL shall be used. A list of exceptions is detailed within the "Licensing Exceptions" section of this document, if one exists.

## Representations, Warranty, and Indemnification

-----

Contributor represents and warrants that the Committed Code does not violate the rights of any person or entity, and that the Contributor has legal authority to enter into this Agreement and legal authority over Contributed Code. Further, Contributor indemnifies Agendaless Consulting against violations.

## Cryptography

-----

Contributor understands that cryptographic code may be subject to government regulations with which Agendaless Consulting and/or entities using Committed Code must comply. Any code which contains any of the items listed below must not be checked-in until Agendaless Consulting staff has been notified and has approved such contribution in writing.

- Cryptographic capabilities or features
- Calls to cryptographic features
- User interface elements which provide context relating to cryptography
- Code which may, under casual inspection, appear to be cryptographic.

## Notices

-----

Contributor confirms that any notices required will be included in any Committed Code.

## Licensing Exceptions

=====

None.

## List of Contributors

=====

The below-signed are contributors to a code repository that is part of the project named "Translationstring". Each below-signed contributor has read, understands and agrees to the terms above in the section within this document entitled "Pylons Project Contributor Agreement" as of the date beside his or her name.

## Contributors

-----

- Chris McDonough, 2011/02/16

- Wichert Akkerman, 2012/02/02

UNRECOGNIZED LICENSE; MD5 sum: 3f2777d4b97006c32953b26ef1f7ac43

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You"

(or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but

not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work

or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

- You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

#### A. HISTORY OF THE SOFTWARE

=====

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see <http://www.cwi.nl>) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see <http://www.cnri.reston.va.us>) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations (now Zope Corporation, see <http://www.zope.com>). In 2001, the Python Software Foundation (PSF, see <http://www.python.org/psf/>) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation is a sponsoring member of the PSF.

All Python releases are Open Source (see <http://www.opensource.org> for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

| Release        | Derived from | Year      | Owner      | GPL-compatible? (1) |
|----------------|--------------|-----------|------------|---------------------|
| 0.9.0 thru 1.2 |              | 1991-1995 | CWI        | yes                 |
| 1.3 thru 1.5.2 | 1.2          | 1995-1999 | CNRI       | yes                 |
| 1.6            | 1.5.2        | 2000      | CNRI       | no                  |
| 2.0            | 1.6          | 2000      | BeOpen.com | no                  |
| 1.6.1          | 1.6          | 2001      | CNRI       | yes (2)             |
| 2.1            | 2.0+1.6.1    | 2001      | PSF        | no                  |
| 2.0.1          | 2.0+1.6.1    | 2001      | PSF        | yes                 |
| 2.1.1          | 2.1+2.0.1    | 2001      | PSF        | yes                 |
| 2.2            | 2.1.1        | 2001      | PSF        | yes                 |
| 2.1.2          | 2.1.1        | 2002      | PSF        | yes                 |
| 2.1.3          |              |           |            |                     |
| 2.1.2          | 2002         | PSF       | yes        |                     |
| 2.2.1          | 2.2          | 2002      | PSF        | yes                 |
| 2.2.2          | 2.2.1        | 2002      | PSF        | yes                 |
| 2.2.3          | 2.2.2        | 2003      | PSF        | yes                 |
| 2.3            | 2.2.2        | 2002-2003 | PSF        | yes                 |
| 2.3.1          | 2.3          | 2002-2003 | PSF        | yes                 |
| 2.3.2          | 2.3.1        | 2002-2003 | PSF        | yes                 |
| 2.3.3          | 2.3.2        | 2002-2003 | PSF        | yes                 |
| 2.3.4          | 2.3.3        | 2004      | PSF        | yes                 |
| 2.3.5          | 2.3.4        | 2005      | PSF        | yes                 |
| 2.4            | 2.3          | 2004      | PSF        | yes                 |
| 2.4.1          | 2.4          | 2005      | PSF        | yes                 |
| 2.4.2          | 2.4.1        | 2005      | PSF        | yes                 |
| 2.4.3          | 2.4.2        | 2006      | PSF        | yes                 |
| 2.4.4          | 2.4.3        | 2006      | PSF        | yes                 |
| 2.5            | 2.4          | 2006      | PSF        | yes                 |
| 2.5.1          | 2.5          | 2007      | PSF        | yes                 |
| 2.5.2          | 2.5.1        | 2008      | PSF        | yes                 |
| 2.5.3          | 2.5.2        | 2008      | PSF        | yes                 |



|       |       |      |     |     |
|-------|-------|------|-----|-----|
| 2.6   | 2.5   | 2008 | PSF | yes |
| 2.6.1 | 2.6   | 2008 | PSF | yes |
| 2.6.2 | 2.6.1 | 2009 | PSF | yes |
| 2.6.3 | 2.6.2 | 2009 | PSF | yes |
| 2.6.4 | 2.6.3 | 2009 | PSF | yes |
| 2.6.5 | 2.6.4 | 2010 | PSF | yes |
| 2.7   | 2.6   | 2010 | PSF | yes |

Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

## B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

=====

### PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

-----

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010 Python Software Foundation; All Rights Reserved" are retained in Python alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on

or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

-----

BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and

otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at <http://www.pythonlabs.com/logos.html> may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

#### CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

-----

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide

license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License

Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the Internet using the following URL: <http://hdl.handle.net/1895.22/1013>".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S.

federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License

Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

#### CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL

DAMAGES OR ANY DAMAGES

WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

UNRECOGNIZED LICENSE; MD5 sum: 1545853f429fe0b615b6e97ded0d7631

#### A. HISTORY OF THE SOFTWARE

=====

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see <http://www.cwi.nl>) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see <http://www.cnri.reston.va.us>) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations (now Zope Corporation, see <http://www.zope.com>). In 2001, the Python Software Foundation (PSF, see <http://www.python.org/psf/>) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation is a sponsoring member of the PSF.

All Python releases are Open Source (see <http://www.opensource.org> for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

| Release        | Derived from | Year      | Owner      | GPL-compatible? (1) |
|----------------|--------------|-----------|------------|---------------------|
| 0.9.0 thru 1.2 |              | 1991-1995 | CWI        | yes                 |
| 1.3 thru 1.5.2 | 1.2          | 1995-1999 | CNRI       | yes                 |
| 1.6            | 1.5.2        | 2000      | CNRI       | no                  |
| 2.0            | 1.6          | 2000      | BeOpen.com | no                  |
| 1.6.1          | 1.6          | 2001      | CNRI       | yes (2)             |
| 2.1            | 2.0+1.6.1    | 2001      | PSF        | no                  |
| 2.0.1          | 2.0+1.6.1    | 2001      | PSF        | yes                 |
| 2.1.1          | 2.1+2.0.1    | 2001      | PSF        | yes                 |
| 2.2            | 2.1.1        | 2001      | PSF        | yes                 |
| 2.1.2          | 2.1.1        | 2002      | PSF        | yes                 |
| 2.1.3          |              |           |            |                     |
| 2.1.2          | 2002         | PSF       | yes        |                     |
| 2.2.1          | 2.2          | 2002      | PSF        | yes                 |
| 2.2.2          | 2.2.1        | 2002      | PSF        | yes                 |
| 2.2.3          | 2.2.2        | 2003      | PSF        | yes                 |
| 2.3            | 2.2.2        | 2002-2003 | PSF        | yes                 |
| 2.3.1          | 2.3          | 2002-2003 | PSF        | yes                 |
| 2.3.2          | 2.3.1        | 2002-2003 | PSF        | yes                 |
| 2.3.3          | 2.3.2        | 2002-2003 | PSF        | yes                 |
| 2.3.4          | 2.3.3        | 2004      | PSF        | yes                 |
| 2.3.5          | 2.3.4        | 2005      | PSF        | yes                 |
| 2.4            | 2.3          | 2004      | PSF        | yes                 |
| 2.4.1          | 2.4          | 2005      | PSF        | yes                 |
| 2.4.2          | 2.4.1        | 2005      | PSF        | yes                 |
| 2.4.3          | 2.4.2        | 2006      | PSF        | yes                 |

|       |       |      |     |     |
|-------|-------|------|-----|-----|
| 2.4.4 | 2.4.3 | 2006 | PSF | yes |
| 2.5   | 2.4   | 2006 | PSF | yes |
| 2.5.1 | 2.5   | 2007 | PSF | yes |
| 2.5.2 | 2.5.1 | 2008 | PSF | yes |
| 2.5.3 | 2.5.2 | 2008 | PSF | yes |
| 2.6   | 2.5   | 2008 | PSF | yes |
| 2.6.1 | 2.6   | 2008 | PSF | yes |
| 2.6.2 | 2.6.1 | 2009 | PSF | yes |
| 2.6.3 | 2.6.2 | 2009 | PSF | yes |
| 2.6.4 | 2.6.3 | 2009 | PSF | yes |
| 2.6.5 | 2.6.4 | 2010 | PSF | yes |
| 3.0   | 2.6   | 2008 | PSF | yes |
| 3.0.1 | 3.0   | 2009 | PSF | yes |
| 3.1   | 3.0.1 | 2009 | PSF | yes |
| 3.1.1 | 3.1   | 2009 | PSF | yes |
| 3.1.2 | 3.1   | 2010 | PSF | yes |
| 3.2   | 3.1   | 2010 | PSF | yes |

#### Footnotes:

- (1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.
- (2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

## B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

=====

### PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

-----

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010 Python Software Foundation; All Rights Reserved" are retained in Python alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

-----

BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1



1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3.  
BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at <http://www.pythonlabs.com/logos.html> may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

-----

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the Internet using the following URL: <http://hdl.handle.net/1895.22/1013>".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

#### CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

-----

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT

OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

UNRECOGNIZED LICENSE; MD5 sum: 8ad4ae5e46a2a3afa27316325386f1cb

UNRECOGNIZED LICENSE; MD5 sum: 45ea32bdebfc2e63ec2b131cd4bc11af

UNRECOGNIZED LICENSE; MD5 sum: c197738b997df284a755c01a887f0ed9

UNRECOGNIZED LICENSE; MD5 sum: 804cd29e64bdef8111fd6b88180e19dc

UNRECOGNIZED LICENSE; MD5 sum: b5e84cb1dddb61a4f3868fc707971432

UNRECOGNIZED LICENSE; MD5 sum: 80d13bfe019695adda309f6a182a2695

UNRECOGNIZED LICENSE; MD5 sum: efc9c2dbdd704e7546d437a42525644c

License

A copyright notice accompanies this license document that identifies the copyright holders.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions in source code must retain the accompanying copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the accompanying copyright notice, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Names of the copyright holders must not be used to endorse or promote products derived from this software without prior written permission from the copyright holders.
4. If any files are modified, you must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

Disclaimer

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

UNRECOGNIZED LICENSE; MD5 sum: c4d2ef9007518f78fd9ae6bb89c51ef7  
UNRECOGNIZED LICENSE; MD5 sum: ef8d703a4d226a68185ce4c5c15d90d7

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN

THE SOFTWARE.

UNRECOGNIZED LICENSE; MD5 sum: 95594488233717b3db751ed31d203303  
MIT License

Copyright (c) 2010-202x The platformdirs developers

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT

OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) Donald Stufft and individual contributors.  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

UNRECOGNIZED LICENSE; MD5 sum: 19a553a1ca654d9e1b3a5c06882c91fa

PK'u5ZtGH#license.dist-0.1.dist-info/METADATAUT ggux]AN0E>rT\*VF]D

(&CjxR5[(Xf\$~"N.vVw5u%Q?8SC'%%q]#n?H]]@6=)))V\_Li

^8r&ZOsyh8P]/Gy\8]C'

c?R:@W5{Sha.PK\*u5ZP%r'!license.dist-0.1.dist-info/RECORDUT ggux}N0I`0W ROonkE0UFqjgd[><i\$

)r^7f!{"3T

]l5w|lQ9!2/7apt81hg\N8|15![A~&x%>PK5d#Ahr6s%license.dist-0.1.dist-info/RECORD.jwsUT

DPmgux%N@EHq()04@:\*`w].9?\'\$mZK'3'

}g7.hl[zMeH4 TQzI9 +g\v7+L? iOU{RC3;>>p\_,7:N-f@]i#

;\_\*bl-X,sZzoI]4a%JQy+KI(:d=^Su`t5U@d"R7PK

d#A~XY@@ license.dist-0.1.dist-info/WHEELUT FDPmguxWheel-Version: 0.1

Packager: bdist\_wheel

Root-Is-Purelib: true

PK

o@licensedist/\_init\_\_.pyUT PBmguxPK'u5ZtGH#license.dist-0.1.dist-

info/METADATAUTguxPK\*u5ZP%r'<license.dist-0.1.dist-info/RECORDUTguxPK5d#Ahr6s%ilicense.dist-0.1.dist-info/RECORD.jwsUTDPuxPK

d#A~XY@@ license.dist-0.1.dist-info/WHEELUTFDPUxPK

o@licensedist/\_init\_\_.pyUTPUxPK

UNRECOGNIZED LICENSE; MD5 sum: c1872b37e59a0ddda1fe4cf8a81192a

UNRECOGNIZED LICENSE; MD5 sum: c628ada423b19fd269a474e38eb93287

Copyright 2012-2021 Eric Larson

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.107 jansson 2.14.1-r0

## 1.107.1 Available under license :

# License

This project is licensed under the MIT license, except where otherwise noted. The full text of the MIT license is included below.

## MIT License

Copyright (c) 2009-2024 Petri Lehtinen <petri@digip.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS

OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## Exceptions

### `src/dtoa.c`

Copyright (c) 1991, 2000, 2001 by Lucent Technologies.

Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice

is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software.

THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHOR NOR LUCENT MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

## 1.108 py3-setuptools 80.9.0-r0

### 1.108.1 Available under license :

The code contained in this directory was automatically generated using the following command:

```
python -m validate_pyproject.pre_compile --output-dir=setuptools/config/_validate_pyproject --enable-plugins
setuptools.distutils --very-verbose -t setuptools=setuptools/config/setuptools.schema.json -t
distutils=setuptools/config/distutils.schema.json
```

Please avoid changing it manually.

You can report issues or suggest changes directly to `validate-pyproject` (or to the relevant plugin repository)

- <https://github.com/abravalheri/validate-pyproject/issues>

\*\*\*

The following files include code from opensource projects (either as direct copies or modified versions):

- `fastjsonschema\_exceptions.py`:
  - project: `fastjsonschema` - licensed under BSD-3-Clause (<https://github.com/horejsek/python-fastjsonschema>)
- `extra\_validations.py` and `format.py`, `error\_reporting.py`:
  - project: `validate-pyproject` - licensed under MPL-2.0 (<https://github.com/abravalheri/validate-pyproject>)

Additionally

the following files are automatically generated by tools provided by the same projects:

- `\_\_init\_\_.py`
- `fastjsonschema\_validations.py`



The relevant copyright notes and licenses are included below.

\*\*\*

`fastjsonschema`

=====

Copyright (c) 2018, Michal Horejsek  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,  
are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this  
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this  
list of conditions and the following disclaimer in the documentation and/or  
other materials provided with the distribution.

Neither the name of the {organization} nor the names of its  
contributors may be used to endorse or promote products derived from  
this software without specific prior written permission.

THIS

SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND  
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED  
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE  
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR  
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES  
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;  
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON  
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT  
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS  
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

\*\*\*

`validate-pyproject`

=====

Mozilla Public License, version 2.0

1. Definitions

1.1. "Contributor"

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2.

"Contributor Version"

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"

means Covered Software of a particular Contributor.

1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"

means

- a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or
- b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. "Executable Form"

means any form of the work other than Source Code Form.

1.7. "Larger Work"

means

a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. "License"

means this document.

1.9. "Licensable"

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

#### 1.10. "Modifications"

means any of the following:

- a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or
- b. any new file in Source Code Form that contains any Covered Software.

#### 1.11. "Patent Claims" of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

#### 1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

#### 1.13. "Source Code Form"

means the form of the work preferred for making modifications.

#### 1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

## 2. License Grants and Conditions

### 2.1.

#### Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- a. under intellectual property rights (other than patent or trademark)  
Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

## 2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

## 2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or
- b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

## 2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if

permitted under the terms of Section 3.3).

## 2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

## 2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

## 2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

## 3. Responsibilities

### 3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

### 3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

### 3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

### 3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

### 3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

## 4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

## 5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days

after You have come back into

compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

## 6. Disclaimer of Warranty

Covered Software is provided under this License on an "as is" basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

## 7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract,

or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

## 8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

## 9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

## 10. Versions of the License

### 10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

### 10.2.

#### Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.



### 10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

### 10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

#### Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <https://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

#### Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

#### MIT License

Copyright (c) 2012 Daniel Holth <dholth@fastmail.fm> and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<https://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

#### 0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

#### 1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

#### 2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

- a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or
- b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

#### 3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

- a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the object code with a copy of the GNU GPL and this license document.

#### 4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the

portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

- a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the Combined Work with a copy of the GNU GPL and this license document.
- c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.
- d) Do one of the following:
  - 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.
  - 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.
- e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

## 5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
- b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

#### 6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

UNRECOGNIZED LICENSE; MD5 sum: 48a0c7f25705055f99a9a86670e60fb8

#### A. HISTORY OF THE SOFTWARE

=====

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see <https://www.cwi.nl>) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see <https://www.cnri.reston.va.us>) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations, which became Zope Corporation. In 2001, the Python Software Foundation (PSF, see <https://www.python.org/psf/>) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation was a sponsoring member of the PSF.

All

Python releases are Open Source (see <https://opensource.org> for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

| Release        | Derived from | Year      | Owner      | GPL-compatible? (1) |
|----------------|--------------|-----------|------------|---------------------|
| 0.9.0 thru 1.2 |              | 1991-1995 | CWI        | yes                 |
| 1.3 thru 1.5.2 | 1.2          | 1995-1999 | CNRI       | yes                 |
| 1.6            | 1.5.2        | 2000      | CNRI       | no                  |
| 2.0            | 1.6          | 2000      | BeOpen.com | no                  |
| 1.6.1          | 1.6          | 2001      | CNRI       | yes (2)             |
| 2.1            | 2.0+1.6.1    | 2001      | PSF        | no                  |
| 2.0.1          | 2.0+1.6.1    | 2001      | PSF        | yes                 |
| 2.1.1          | 2.1+2.0.1    | 2001      | PSF        | yes                 |
| 2.1.2          | 2.1.1        | 2002      | PSF        | yes                 |
| 2.1.3          | 2.1.2        | 2002      | PSF        | yes                 |
| 2.2 and above  | 2.1.1        | 2001-now  |            |                     |
| PSF            | yes          |           |            |                     |

Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

## B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

Python software and documentation are licensed under the Python Software Foundation License Version 2.

Starting with Python 3.8.6, examples, recipes, and other code in the documentation are dual licensed under the PSF License Version 2 and the Zero-Clause BSD license.

Some software incorporated into Python is under different licenses. The licenses are listed with code falling under that license.

### PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.
2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023 Python Software Foundation; All Rights Reserved" are retained in Python alone or in any derivative version prepared by Licensee.
3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.
4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS

FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

#### BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

#### BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE



WILL NOT  
INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at <http://www.pythonlabs.com/logos.html> may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

#### CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

-----

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the

quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the internet using the following URL: <http://hdl.handle.net/1895.22/1013>".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying,

installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

#### CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

-----

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER

RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

#### ZERO-CLAUSE BSD LICENSE FOR CODE IN THE PYTHON DOCUMENTATION

-----

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

UNRECOGNIZED LICENSE; MD5 sum: 525adedaf9a94edbb3e245cdd7b0f448

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

#### END OF TERMS AND CONDITIONS

This software is made available under the terms of *\*either\** of the licenses found in LICENSE.APACHE or LICENSE.BSD. Contributions to this software is made under the terms of *\*both\** these licenses.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in

all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN

THE SOFTWARE.

MIT License

Copyright (c) 2021 Taneli Hukkinen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2010-202x The platformdirs developers

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,



FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT

OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) Donald Stuft and individual contributors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2012 Erik Rose

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE

OR THE USE OR OTHER DEALINGS IN THE  
SOFTWARE.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute

copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following

boilerplate notice, with the fields enclosed by brackets "[ ]"  
replaced with your own identifying information. (Don't include  
the brackets!) The text should be enclosed in the appropriate  
comment syntax for the file format. We also recommend that a  
file or class name and description of purpose be included on the  
same "printed page" as the copyright notice for easier  
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

The following files include code from opensource projects  
(either as direct copies or modified versions):

- `setuptools.schema.json`, `distutils.schema.json`:  
- project: `validate-pyproject` - licensed under MPL-2.0  
(<https://github.com/abravalheri/validate-pyproject>):

This Source Code Form is subject to the terms of the Mozilla Public  
License, v. 2.0. If a copy of the MPL was not distributed with this file,  
You can obtain one at <https://mozilla.org/MPL/2.0/>.

This is the MIT license: <http://www.opensource.org/licenses/mit-license.php>

Copyright (c) Alex Grnholm

Permission is hereby granted, free of charge, to any person obtaining a copy of this  
software and associated documentation files (the "Software"), to deal in the Software  
without restriction, including without limitation the rights to use, copy, modify, merge,  
publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons  
to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or  
substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,  
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A  
PARTICULAR  
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS

BE LIABLE  
FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,  
TORT OR  
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR  
OTHER  
DEALINGS IN THE SOFTWARE.

# 1.109 libcap 2.76-r0

## 1.109.1 Available under license :

/\* SPDX-License-Identifier: BSD-3-Clause OR GPL-2.0-only \*/

Unless otherwise *\*explicitly\** stated, the following text describes the  
licensed conditions under which the contents of this libcap/cap release  
may be used and distributed.

The licensed conditions are one or the other of these two Licenses:

- BSD 3-clause
- GPL v2.0

-----  
BSD 3-clause:  
-----

Redistribution and use in source and binary forms of libcap/cap, with  
or without modification, are permitted provided that the following  
conditions are met:

1. Redistributions of source code must retain any existing copyright  
notice, and this entire permission notice in its entirety,  
including the disclaimer of warranties.
2. Redistributions in binary form must reproduce all prior and current  
copyright notices, this list of conditions, and the following  
disclaimer in the documentation and/or other materials provided  
with the distribution.
3.  
The name of any author may not be used to endorse or promote  
products derived from this software without their specific prior  
written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED  
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF  
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.  
IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT,

INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-----  
GPL v2.0:  
-----

ALTERNATIVELY, this product may be distributed under the terms of the GNU General Public License (v2.0 - see below), in which case the provisions of the GNU GPL are required INSTEAD OF the above restrictions. (This clause is necessary due to a potential conflict between the GNU GPL and the restrictions contained in a BSD-style copyright.)

-----  
Full text of gpl-2.0.txt:  
-----

## GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for



this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in

the term "modification".) Each licensee is addressed as "you".

#### Activities

other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If

identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you;  
rather, the intent is to  
exercise the right to control the distribution of derivative or  
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete

machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include

anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by

all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author

to ask for permission. For software which is copyrighted by the Free Software

Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

## NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL

ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey

the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify

it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information  
on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which
makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

```
/* SPDX-License-Identifier: BSD-3-Clause OR GPL-2.0-only */
```

Unless otherwise *\*explicitly\** stated, the following text describes the

licensed conditions under which the contents of this libcap/psx release may be used and distributed.

The licensed conditions are one or the other of these two Licenses:

- BSD 3-clause
- GPL v2.0

-----  
BSD 3-clause:  
-----

Redistribution and use in source and binary forms of libcap/psx, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce all prior and current copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3.  
The name of any author may not be used to endorse or promote products derived from this software without their specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-----  
GPL v2.0:  
-----

ALTERNATIVELY, this product may be distributed under the



terms of the  
GNU General Public License (v2.0 - see below), in which case the  
provisions of the GNU GPL are required INSTEAD OF the above  
restrictions. (This clause is necessary due to a potential conflict  
between the GNU GPL and the restrictions contained in a BSD-style  
copyright.)

-----  
Full text of gpl-2.0.txt:  
-----

## GNU GENERAL PUBLIC LICENSE

### Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your  
freedom to share and change it. By contrast, the GNU General Public  
License is intended to guarantee your freedom to share and change free  
software--to make sure the software is free for all its users.

This  
General Public License applies to most of the Free Software  
Foundation's software and to any other program whose authors commit to  
using it. (Some other Free Software Foundation software is covered by  
the GNU Lesser General Public License instead.) You can apply it to  
your programs, too.

When we speak of free software, we are referring to freedom, not  
price. Our General Public Licenses are designed to make sure that you  
have the freedom to distribute copies of free software (and charge for  
this service if you wish), that you receive source code or can get it  
if you want it, that you can change the software or use pieces of it  
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid  
anyone to deny you these rights or to ask you to surrender the rights.  
These restrictions translate to certain responsibilities for you if you  
distribute copies of the software, or if you modify it.

For example, if  
you distribute copies of such a program, whether  
gratis or for a fee, you must give the recipients all the rights that  
you have. You must make sure that they, too, receive or can get the

source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

#### Activities

other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate

copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or

collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete

machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program

except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the

integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software

Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED

OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL

ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey

the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information  
on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this  
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate  
parts of the General Public License. Of course, the commands you use may  
be called something other than `show w' and `show c'; they could even be  
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your  
school, if any, to sign a "copyright disclaimer" for the program, if  
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which  
makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into  
proprietary programs. If your program is a subroutine library, you may  
consider it more useful to permit linking proprietary applications with the  
library. If this is what you want to do, use the GNU Lesser General  
Public License instead of this License.

/\* SPDX-License-Identifier: BSD-3-Clause OR GPL-2.0-only \*/

Unless otherwise *\*explicitly\** stated, the following text describes the  
licensed conditions under which the contents of this libcap release  
may be used and distributed.

The licensed conditions are one or the other of these two Licenses:

- BSD 3-clause
- GPL v2.0

-----  
BSD 3-clause:

-----

Redistribution and use in source and binary forms of libcap, with



or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce all prior and current copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name  
of any author may not be used to endorse or promote  
products derived from this software without their specific prior  
written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-----  
GPL v2.0:  
-----

ALTERNATIVELY, this product may be distributed under the terms of the  
GNU General Public License (v2.0 - see below), in which case the provisions of the GNU GPL are required INSTEAD OF the above restrictions.

-----  
Full text of gpl-2.0.txt:  
-----

GNU GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using

it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free

program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition,

mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your

cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates

the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if



necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public

License does not permit incorporating your program into  
proprietary programs. If your program is a subroutine library, you may  
consider it more useful to permit linking proprietary applications with the  
library. If this is what you want to do, use the GNU Lesser General  
Public License instead of this License.

/\* SPDX-License-Identifier: BSD-3-Clause OR LGPL-2.0-or-later \*/

Unless otherwise *\*explicitly\** stated the following text describes the  
licensed conditions under which the contents of this module release  
may be distributed:

-----  
Redistribution and use in source and binary forms of this module, with  
or without modification, are permitted provided that the following  
conditions are met:

1. Redistributions of source code must retain any existing copyright  
notice, and this entire permission notice in its entirety,  
including the disclaimer of warranties.
2. Redistributions in binary form must reproduce all prior and current  
copyright notices, this list of conditions, and the following  
disclaimer in the documentation and/or other materials provided  
with the distribution.
3. The name of any author may not be used to endorse or promote  
products derived from this software without their specific prior  
written  
permission.

ALTERNATIVELY, this product may be distributed under the terms of the  
GNU Library General Public License, in which case the provisions of  
the GNU LGPL are required INSTEAD OF the above restrictions. (This  
clause is necessary due to a potential conflict between the GNU LGPL  
and the restrictions contained in a BSD-style copyright.)

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED  
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF  
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-----

## 1.110 openssl 3.5.4-r0

### 1.110.1 Available under license :

This software is copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software; you can redistribute it and/or modify it under the same terms as the Perl 5 programming language system itself.

Terms of the Perl programming language system itself

- a) the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version, or
- b) the "Artistic License"

--- The GNU General Public License, Version 1, February 1989 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The GNU General Public License, Version 1, February 1989

GNU GENERAL PUBLIC LICENSE  
Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.  
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our General Public

License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. The General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each

licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use

in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

- a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,
- b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,
- c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to

address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it

free software which everyone can redistribute and change under these terms.

To do so,  
attach the following notices to the program. It is safest to  
attach them to the start of each source file to most effectively convey  
the exclusion of warranty; and each file should have at least the  
"copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify  
it under the terms of the GNU General Public License as published by  
the Free Software Foundation; either version 1, or (at your option)  
any later version.

This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.

You should have received a copy of the GNU General Public License  
along with this program; if not, write to the Free  
Software  
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this  
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19xx name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the  
appropriate parts of the General Public License. Of course, the  
commands you use may be called something other than `show w' and `show  
c'; they could even be mouse-clicks or menu items--whatever suits your  
program.

You should also get your employer (if you work as a programmer) or your  
school, if any, to sign a "copyright disclaimer" for the program, if  
necessary. Here a sample; alter

the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (a program to direct compilers to make passes at assemblers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

That's all there is to it!

--- The Artistic License 1.0 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The Artistic License 1.0

The Artistic License

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

- "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.
- "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.
- "Copyright Holder" is whoever is named in the copyright or copyrights for the package.
- "You" is you, if you're thinking about copying or distributing this Package.
- "Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)
- "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that



recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:
  - a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.
  - b) use the modified Package only within your corporation or organization.
  - c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.
  - d) make other distribution arrangements with the Copyright Holder.
4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:
  - a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.
  - b) accompany the distribution with the machine-readable source of the Package with your modifications.
  - c) accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard names,  
and clearly documenting the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.

7. C or perl subroutines supplied by you and linked into this Package shall not be considered part of this Package.

8. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

9. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

Apache License  
Version 2.0, January 2004  
<https://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the

outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,  
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

## 1.111 pyparsing 3.2.3

### 1.111.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.112 ltrace 0.7.3-r6

## 1.112.1 Available under license :

This is the Debian GNU/Linux's prepackaged version of the Dynamic Library Tracer ``ltrace".

It was downloaded from <http://www.ltrace.org/>

### Copyrights

-----

Copyright (C) 1997-2009 Juan Céspedes <cespedes@debian.org>

ARMLinux port: Copyright (C) 1998 Pat Beirne <pbeirne@home.com>

m68k port: Copyright (C) 1998 Roman Hodek <Roman.Hodek@informatik.uni-erlangen.de>

Misc fixes: Copyright (C) 1999 Morten Eriksen <mortene@sim.no>

s390 port: Copyright (C) 2001 IBM Poughkeepsie, IBM Cororation <slate@us.ibm.com>

ELF hacking: Copyright (C) 1999 Silvio Cesare <silvio@big.net.au>

PowerPC port: Copyright (C) 2001-2002 Anton Blanchard <anton@samba.org>

SPARC port: Copyright (C) 1999 Jakub Jelinek <jakub@redhat.com>

C++ demangle: Copyright 1989-1997 Free Software Foundation, Inc.

### License

-----

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

A copy of the GNU General Public License is available as `/usr/share/common-licenses/GPL` in the Debian GNU/Linux distribution or on the World Wide Web at `'http://www.gnu.org/copyleft/gpl.html'`.

You can also obtain it by writing to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software



patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in

whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can

be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

### 3. You may copy and distribute the Program

(or a work based on it,

under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based

on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding

those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE

## POSSIBILITY OF SUCH DAMAGES.

### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be

mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

## 1.113 libnetfilter-conntrack 1.1.0-r0

### 1.113.1 Available under license :

GNU GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
675 Mass Ave, Cambridge, MA 02139, USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it

in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".



Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably

considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary

form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6.

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then

the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free

Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

## NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of

the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.

You should have received a copy of the GNU General Public License  
along with this program; if not, write to the Free Software  
Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this  
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c'  
should show the appropriate  
parts of the General Public License. Of course, the commands you use may  
be called something other than `show w' and `show c'; they could even be  
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your  
school, if any, to sign a "copyright disclaimer" for the program, if  
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into  
proprietary programs. If your program is a subroutine library, you may  
consider it more useful to permit linking proprietary applications with the  
library. If this is what you want to do, use the GNU Library General  
Public License instead of this License.

## 1.114 truststore 0.10.0

## 1.114.1 Available under license :

MIT

The MIT License (MIT)

Copyright (c) 2022 Seth Michael Larson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.115 json-c 0.18-r1

### 1.115.1 Available under license :

This work was packaged for Debian by:

William Dauchy <wdauchy@gmail.com> on Tue, 16 Mar 2010 15:45:59 +0100

It was downloaded from:

<http://projects.snarc.org/libjson/>

Upstream Author(s):

Vincent Hanquez <tab@snarc.org>

Copyright:

Copyright (C) 2009-2010 Vincent Hanquez

License:

libjson is covered by the LGPLv2 license, or at your option the LGPLv3 license

The Debian packaging is:

Copyright (C) 2010 William Dauchy <[wdauchy@gmail.com](mailto:wdauchy@gmail.com)>

and is licensed under the LGPL version 3,  
see "/usr/share/common-licenses/LGPL".

# 1.116 gettext 0.24.1-r0

## 1.116.1 Available under license :

The MIT License (MIT)

Copyright (c) 2017 Maxim Sokolov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR

IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.

<<https://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble



The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but

we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the

existence of

any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination

fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the

users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL  
PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's

complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based

on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or

linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License.

Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6.

Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding

machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

- b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License.



If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by

the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

##### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively

convey the exclusion of warranty; and  
each file should have at least the  
"copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or  
modify it under the terms of the GNU Lesser General Public  
License as published by the Free Software Foundation; either  
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU  
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public  
License along with this library; if not, see <<https://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

You should also get your employer  
(if you work as a programmer) or your  
school, if any, to sign a "copyright disclaimer" for the library, if  
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the  
library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Moe Ghoul>, 1 April 1990  
Moe Ghoul, President of Vice

That's all there is to it!  
The gettext-runtime package is partially under the LGPL and partially under  
the GPL.

The following parts are under the LGPL, see file intl/COPYING.LIB:  
- the libintl and libasprintf libraries and their header files,  
- the libintl.jar Java library,  
- the GNU.Gettext.dll C# library,  
- the gettext.sh shells script function library.

The following parts are under the GPL, see file COPYING in the toplevel  
directory:  
- the \_programs\_ gettext, ngettext, envsubst,  
- the documentation.

## GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them

with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

^L

Finally,  
software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it

becomes

a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

^L

## GNU LESSER GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly

into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or

table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

^L

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.



4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the

complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

^L

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

^L

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited

by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the

Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

^L

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the

original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

### 13. The Free Software Foundation

may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

^L

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE

LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

^L

#### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or

your  
school, if any, to sign a "copyright disclaimer" for the library, if  
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the  
library `Frob' (a library for tweaking knobs) written by James  
Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!  
COPYRIGHT AND PERMISSION NOTICE (ICU 58 and later)

Copyright 1991-2019 Unicode, Inc. All rights reserved.  
Distributed under the Terms of Use in <https://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining  
a copy of the Unicode data files and any associated documentation  
(the "Data Files") or Unicode software and any associated documentation  
(the "Software") to deal in the Data Files or Software  
without restriction, including without limitation the rights to use,  
copy, modify, merge, publish, distribute, and/or sell copies of  
the Data Files or Software, and to permit persons to whom the Data Files  
or Software are furnished to do so, provided that either  
(a) this copyright and permission notice appear with all copies  
of the Data Files or Software, or  
(b) this copyright and permission notice appear in associated  
Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF  
ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT  
NOT LIMITED TO THE  
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND  
NONINFRINGEMENT OF THIRD PARTY RIGHTS.  
IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS  
NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL  
DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,  
DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER  
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR  
PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder  
shall not be used in advertising or otherwise to promote the sale,  
use or other dealings in these Data Files or Software without prior  
written authorization of the copyright holder.

-----

## Third-Party Software Licenses

This section contains third-party software notices and/or additional terms for licensed third-party software components included within ICU libraries.

### 1. ICU License - ICU 1.8.1 to ICU 57.1

#### COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2016 International Business Machines Corporation and others  
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.

### 2. Chinese/Japanese Word Break Dictionary Data (cjdict.txt)

# The Google Chrome software developed by Google is licensed under  
# the BSD license. Other software included in this distribution is



```
# provided under other licenses, as set forth below.
#
# The BSD License
# http://opensource.org/licenses/bsd-license.php
# Copyright (C) 2006-2008, Google Inc.
#
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are met:
#
# Redistributions of source code must retain the above copyright notice,
# this list of conditions and the following disclaimer.
# Redistributions in binary form must reproduce the above
# copyright notice, this list of conditions and the following
# disclaimer in the documentation and/or other materials provided with
# the distribution.
# Neither the name of Google Inc. nor the names of its
# contributors may be used to endorse or promote products derived from
# this software without specific prior written permission.
#
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
# CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
# INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
# MERCHANTABILITY AND FITNESS FOR A PARTICULAR
# PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
# LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
# CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
# SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
# BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
# LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
# SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
#
# The word list in cjdict.txt are generated by combining three word lists
# listed below with further processing for compound word breaking. The
# frequency is generated with an iterative training against Google web
# corpora.
#
# * Libtabe (Chinese)
#   - https://sourceforge.net/project/?group\_id=1519
#   - Its license terms and conditions are shown below.
#
# * IPADIC (Japanese)
#   - http://chasen.aist-nara.ac.jp/chasen/distribution.html
```

```

# - Its license terms and conditions are shown below.
#
# -----COPYING.libtabe ---- BEGIN-----
#
# /*
#  * Copyright (c) 1999 TaBE Project.
#  * Copyright (c) 1999 Pai-Hsiang Hsiao.
#  * All rights reserved.
#  *
#  * Redistribution and use in source and binary forms, with or without
#  * modification, are permitted provided that the following conditions
#  * are met:
#  *
#  * . Redistributions of source code must retain the above copyright
#  * notice, this list of conditions and the following disclaimer.
#  * . Redistributions in binary form must reproduce the above copyright
#  * notice, this list of conditions and the following disclaimer in
#  * the documentation and/or other materials provided with the
#  * distribution.
#  * . Neither the name of the TaBE Project nor the names of its
#  * contributors may be used to endorse or promote products derived
#  * from this software
#  * without specific prior written permission.
#  *
#  * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
#  * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
#  * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
#  * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
#  * REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
#  * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
#  * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
#  * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
#  * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
#  * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
#  * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
#  * OF THE POSSIBILITY OF SUCH DAMAGE.
#  */
#
# /*
#  * Copyright (c) 1999 Computer Systems and Communication Lab,
#  *
#  *      Institute
#  * of Information Science, Academia
#  *
#  *      Sinica. All rights reserved.
#  *
#  *
#  * Redistribution and use in source and binary forms, with or without
#  * modification, are permitted provided that the following conditions
#  * are met:
#  *
#  *

```

```

# * . Redistributions of source code must retain the above copyright
# * notice, this list of conditions and the following disclaimer.
# * . Redistributions in binary form must reproduce the above copyright
# * notice, this list of conditions and the following disclaimer in
# * the documentation and/or other materials provided with the
# * distribution.
# * . Neither the name of the Computer Systems and Communication Lab
# * nor the names of its contributors may be used to endorse or
# * promote products derived from this software without specific
# * prior written permission.
# *
# * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# * "AS IS" AND
# * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# * REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
# * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# * OF THE POSSIBILITY OF SUCH DAMAGE.
# */
#
# Copyright 1996 Chih-Hao Tsai @ Beckman Institute,
# University of Illinois
# c-tsai4@uiuc.edu http://casper.beckman.uiuc.edu/~c-tsai4
#
# -----COPYING.libtabe-----END-----
#
#
# -----COPYING.ipadic-----BEGIN-----
#
# Copyright 2000, 2001, 2002, 2003 Nara Institute of Science
# and Technology. All Rights Reserved.
#
# Use, reproduction, and distribution of this software is permitted.
# Any copy of this software, whether in its original form or modified,
# must include both the above copyright notice and the following
# paragraphs.
#
# Nara Institute of Science and Technology (NAIST),
# the copyright holders, disclaims all warranties with regard to this
# software, including all implied warranties of merchantability and
# fitness, in no event shall NAIST be liable for
# any special, indirect or consequential damages or any damages

```

# whatsoever resulting from loss of use, data or profits, whether in an  
# action of contract, negligence or other tortuous action, arising out  
# of or in connection with the use or performance of this software.  
#  
# A large portion of the dictionary entries  
# originate from ICOT  
Free Software. The following conditions for ICOT  
# Free Software applies to the current dictionary as well.  
#  
# Each User may also freely distribute the Program, whether in its  
# original form or modified, to any third party or parties, PROVIDED  
# that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear  
# on, or be attached to, the Program, which is distributed substantially  
# in the same form as set out herein and that such intended  
# distribution, if actually made, will neither violate or otherwise  
# contravene any of the laws and regulations of the countries having  
# jurisdiction over the User or the intended distribution itself.  
#  
# NO WARRANTY  
#  
# The program was produced on an experimental basis in the course of the  
# research and development conducted during the project and is provided  
# to users as so produced on an experimental basis. Accordingly, the  
# program is provided without any warranty whatsoever, whether express,  
# implied,  
statutory or otherwise. The term "warranty" used herein  
# includes, but is not limited to, any warranty of the quality,  
# performance, merchantability and fitness for a particular purpose of  
# the program and the nonexistence of any infringement or violation of  
# any right of any third party.  
#  
# Each user of the program will agree and understand, and be deemed to  
# have agreed and understood, that there is no warranty whatsoever for  
# the program and, accordingly, the entire risk arising from or  
# otherwise connected with the program is assumed by the user.  
#  
# Therefore, neither ICOT, the copyright holder, or any other  
# organization that participated in or was otherwise related to the  
# development of the program and their respective officials, directors,  
# officers and other employees shall be held liable for any and all  
# damages, including, without limitation, general, special, incidental  
# and consequential damages, arising out of or otherwise in  
connection  
# with the use or inability to use the program or any product, material  
# or result produced or otherwise obtained by using the program,  
# regardless of whether they have been advised of, or otherwise had  
# knowledge of, the possibility of such damages at any time during the  
# project or thereafter. Each user will be deemed to have agreed to the

```
# foregoing by his or her commencement of use of the program. The term
# "use" as used herein includes, but is not limited to, the use,
# modification, copying and distribution of the program and the
# production of secondary products from the program.
#
# In the case where the program, whether in its original form or
# modified, was distributed or delivered to or received by a user from
# any person, organization or entity other than ICOT, unless it makes or
# grants independently of ICOT any specific warranty to the user in
# writing, such person, organization or entity, will also be exempted
# from
and not be held liable to the user for any such damages as noted
# above as far as the program is concerned.
#
# -----COPYING.ipadic-----END-----
```

### 3. Lao Word Break Dictionary Data (laodict.txt)

```
# Copyright (c) 2013 International Business Machines Corporation
# and others. All Rights Reserved.
#
# Project: http://code.google.com/p/lao-dictionary/
# Dictionary: http://lao-dictionary.googlecode.com/git/Lao-Dictionary.txt
# License: http://lao-dictionary.googlecode.com/git/Lao-Dictionary-LICENSE.txt
# (copied below)
#
# This file is derived from the above dictionary, with slight
# modifications.
# -----
# Copyright (C) 2013 Brian Eugene Wilson, Robert Martin Campbell.
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification,
# are permitted provided that the following conditions are met:
#
#
# Redistributions of source code must retain the above copyright notice, this
# list of conditions and the following disclaimer. Redistributions in
# binary form must reproduce the above copyright notice, this list of
# conditions and the following disclaimer in the documentation and/or
# other materials provided with the distribution.
#
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
```

# COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,  
# INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES  
# (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR  
# SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
# HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,  
# STRICT LIABILITY, OR  
# TORT (INCLUDING NEGLIGENCE OR OTHERWISE)  
# ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED  
# OF THE POSSIBILITY OF SUCH DAMAGE.  
# -----

#### 4. Burmese Word Break Dictionary Data (burmesedict.txt)

# Copyright (c) 2014 International Business Machines Corporation  
# and others. All Rights Reserved.  
#  
# This list is part of a project hosted at:  
# [github.com/kanyawtech/myanmar-karen-word-lists](https://github.com/kanyawtech/myanmar-karen-word-lists)  
#  
# -----  
# Copyright (c) 2013, LeRoy Benjamin Sharon  
# All rights reserved.  
#  
# Redistribution and use in source and binary forms, with or without  
# modification, are permitted provided that the following conditions  
# are met: Redistributions of source code must retain the above  
# copyright notice, this list of conditions and the following  
# disclaimer. Redistributions in binary form must reproduce the  
# above  
# copyright notice, this list of conditions and the following  
# disclaimer in the documentation and/or other materials provided  
# with the distribution.  
#  
# Neither the name Myanmar Karen Word Lists, nor the names of its  
# contributors may be used to endorse or promote products derived  
# from this software without specific prior written permission.  
#  
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND  
# CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,  
# INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF  
# MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE  
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS  
# BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,  
# EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED  
# TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,  
# DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON  
# ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,

STRICT LIABILITY, OR  
# TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF  
# THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF  
# SUCH DAMAGE.  
# -----

## 5. Time Zone Database

ICU uses the public domain data and code derived from Time Zone Database for its time zone support. The ownership of the TZ database is explained in BCP 175: Procedure for Maintaining the Time Zone Database section 7.

### # 7. Database Ownership

#  
# The TZ database itself is not an IETF Contribution or an IETF  
# document. Rather it is a pre-existing and regularly updated work  
# that is in the public domain, and is intended to remain in the  
# public domain. Therefore, BCPs 78 [RFC5378] and 79 [RFC3979] do  
# not apply to the TZ Database or contributions that individuals make  
# to it. Should any claims be made and substantiated against the TZ  
# Database, the organization that is providing  
the IANA  
# Considerations defined in this RFC, under the memorandum of  
# understanding with the IETF, currently ICANN, may act in accordance  
# with all competent court orders. No ownership claims will be made  
# by ICANN or the IETF Trust on the database or the code. Any person  
# making a contribution to the database or code waives all rights to  
# future claims in that contribution or in the TZ Database.

## 6. Google double-conversion

Copyright 2006-2011, the V8 project authors. All rights reserved.  
Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions are  
met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#### GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<https://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive



or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1)

assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand

ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS

### 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees"

and

"recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

#### 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The

"System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that

Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

## 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of

your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

### 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

### 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

### 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section

7. This requirement modifies the requirement in section 4 to

"keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

## 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical

medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

## 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option

remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty  
or limiting liability differently from the  
terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or  
author attributions in that material or in the Appropriate Legal  
Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or  
requiring that modified versions of such material be marked in  
reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or  
authors of the material; or
- e) Declining to grant rights under trademark law for use of some  
trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that  
material by anyone who conveys the material (or modified versions of  
it) with contractual assumptions of liability to the recipient, for  
any liability that these contractual assumptions directly impose on  
those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating



where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

#### 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

#### 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

## 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

## 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this

License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting

any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

#### 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

#### 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

#### 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

#### 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

#### 16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### 17.

Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>

This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<https://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <<https://www.gnu.org/philosophy/why-not-lgpl.html>>.

/\* Extract the last component (base name) of a file name.

Copyright (C) 1998, 2001, 2003-2006, 2009-2025 Free Software Foundation, Inc.

This file is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This file is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>. \*/

```
#ifndef _BASENAME_LGPL_H
#define _BASENAME_LGPL_H
```

```
/* This file uses _GL_ATTRIBUTE_PURE. */
#if !_GL_CONFIG_H_INCLUDED
#error "Please include config.h first."
#endif
```

```
#include <stddef.h>
```

```
#ifndef
DOUBLE_SLASH_IS_DISTINCT_ROOT
# define DOUBLE_SLASH_IS_DISTINCT_ROOT 0
#endif
```

```
#ifdef __cplusplus
extern "C" {
#endif
```

/\* Return the address of the last file name component of FILENAME.  
If FILENAME has some trailing slash(es), they are considered to be part of the last component.  
If FILENAME has no relative file name components because it is a file system root, return the empty string.  
Examples:

| FILENAME     | RESULT  |
|--------------|---------|
| "foo.c"      | "foo.c" |
| "foo/bar.c"  | "bar.c" |
| "/foo/bar.c" | "bar.c" |

```

"foo/bar/"  "bar/"
"foo/bar//" "bar//"
"/"        ""
"//"       ""
""         ""

```

The return value is a tail of the given FILENAME; do NOT free() it! \*/

/\* This function was traditionally called 'basename', but we avoid this function name because

\* Various platforms have different functions in their libc.

In particular, the glibc basename(),

defined in <string.h>, does

not consider trailing slashes to be part of the component:

```

FILENAME  RESULT
"foo/bar/" ""
"foo/bar//" ""

```

\* The 'basename' command eliminates trailing slashes and for a root produces a non-empty result:

```

FILENAME  RESULT
"foo/bar/" "bar"
"foo/bar//" "bar"
"/"        "/"
"//"       "/"

```

\*/

extern char \*last\_component (char const \*filename) \_GL\_ATTRIBUTE\_PURE;

/\* Return the length of the basename FILENAME.

Typically FILENAME is the value returned by base\_name or last\_component.

Act like strlen (FILENAME), except omit all trailing slashes. \*/

extern size\_t base\_len (char const \*filename) \_GL\_ATTRIBUTE\_PURE;

#ifdef \_\_cplusplus

} /\* extern "C" \*/

#endif

#endif /\* \_BASENAME\_LGPL\_H \*/

The libasprintf package is under the LGPL, see file COPYING.LIB.

This subpackage is under the GPL, see file COPYING in the toplevel directory.

The MIT License (MIT)

Copyright (c) 2017 Max Brunsfeld

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is



furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR

IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except where otherwise noted in the source code (e.g. the files hash.c, list.c and the trio files, which are covered by a similar licence but with different Copyright notices) all the files are:

Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE

AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2018-2024 Max Brunsfeld

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT

OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS

IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014 Max Brunsfeld

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.117 liburcu 0.15.2-r0

### 1.117.1 Available under license :

As a special exception, the Free Software Foundation gives unlimited permission to copy, distribute and modify the configure scripts that are the output of Autoconf. You need not follow the terms of the GNU General Public License when using or distributing such scripts, even though portions of the text of Autoconf appear in them. The GNU General Public License (GPL) does govern all other use of the material that constitutes the Autoconf program.

Certain portions of the Autoconf source text are designed to be copied (in certain cases, depending on the input) into the output of Autoconf. We call these the "data" portions. The rest of the Autoconf source text consists of comments plus executable code that decides which of the data portions to output in any given case. We call these comments and executable code the "non-data" portions. Autoconf never copies any of the non-data portions into its output.

This special exception to the GPL applies to versions of Autoconf released by the Free Software Foundation. When you make and distribute a modified version of Autoconf, you may extend this special exception to the GPL to apply to your modified version as well, \*unless\* your modified version has the potential to copy into its output some of the text that was the non-data portion of the version that you started with. (In other words, unless your change moves or copies text from the non-data portions to the data portions.) If your modification has such potential, you must delete any notice of this special exception to the GPL from your modified version.

Copyright (c) ...

THIS MATERIAL IS PROVIDED AS IS, WITH ABSOLUTELY NO WARRANTY EXPRESSED OR IMPLIED. ANY USE IS AT YOUR OWN RISK.

Permission is hereby granted to use or copy this program for any purpose, provided the above notices are retained on all copies. Permission to modify the code and to distribute modified code is granted, provided the above notices are retained, and a notice that the code was modified is included with the above copyright notice.

A few files have other copyright holders.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom

to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for

each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under

the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its

scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work,

complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense

or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the

Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use

of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the

Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY

COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the program's name and an idea of what it does. Copyright (C) yyyy name of author

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA. Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:



Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989 Ty Coon, President of Vice  
GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

## Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully

about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you.

You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the

library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software

library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square

roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work

based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License.

(If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative

work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely

defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself

accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software

distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT

WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply

#### These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the library's name and an idea of what it does.

Copyright (C) year name of author

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in  
the library `Frob' (a library for tweaking knobs) written  
by James Random Hacker.

signature of Ty Coon, 1 April 1990

Ty Coon, President of Vice

That's all there is to it!

<!--

SPDX-FileCopyrightText: 2009 Mathieu Desnoyers <mathieu.desnoyers@efficios.com>

SPDX-License-Identifier: CC-BY-4.0

-->

# Userspace-RCU LGPL 2.1+ re-licensing

Mathieu Desnoyers



May 13th, 2009

IBM Corporation allowed LGPLv2.1+ licensing of their contribution to the userspace RCU library in a patch submitted on May 8, 2009 from Paul E. McKenney and reviewed by Steven L. Bennett:

<https://lists.lttng.org/pipermail/lttng-dev/2009-May/012835.html>

I (Mathieu Desnoyers) re-implemented ACCESS\_ONCE(), likely(), unlikely() and barrier() from scratch without reference to the original code.

commit id : 2dc5fa0f7cfbfb0a64a7a67b39626650e863f16a

Bert Wesarg <bert.wesarg@googlemail.com> approved LGPL re-licensing of his patch in an email dated May 13, 2009 :

<http://lkml.org/lkml/2009/5/13/16>

xchg() primitives has been rewritten from a MIT-licensed cmpxchg for Intel and powerpc. They are MIT-licensed and therefore usable in LGPL code. This cmpxchg code was obtained from the atomic\_ops project:

[http://www.hpl.hp.com/research/linux/atomic\\_ops/](http://www.hpl.hp.com/research/linux/atomic_ops/)

I (Mathieu Desnoyers) wrote the remainder of the code.

The license for the library files in this project was therefore changed to LGPLv2.1 on May 13, 2009, as detailed in LICENSE.  
MIT License

Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

As a special exception, the respective Autoconf Macro's copyright owner gives unlimited permission to copy, distribute and modify the configure scripts that are the output of Autoconf when processing the Macro. You need not follow the terms of the GNU General Public License when using or distributing such scripts, even though portions of the text of the Macro appear in them. The GNU General Public License (GPL) does govern all other use of the material that constitutes the Autoconf Macro.

This special exception to the GPL applies to versions of the Autoconf Macro released by the Autoconf Archive. When you make and distribute a modified version of the Autoconf Macro, you may extend this special exception to the GPL to apply to your modified version as well.

Creative Commons Attribution 4.0 International

Creative Commons Corporation (Creative Commons) is not a law firm and does not provide legal services or legal advice. Distribution of Creative Commons public licenses does not create a lawyer-client or other relationship. Creative Commons makes its licenses and related information available on an as-is basis. Creative Commons gives no warranties regarding its licenses, any material licensed under their terms and conditions, or any related information. Creative Commons disclaims all liability for damages resulting from their use to the fullest extent possible.

#### Using Creative Commons Public Licenses

Creative Commons public licenses provide a standard set of terms and conditions that creators and other rights holders may use to share original works of authorship and other material subject to copyright and certain other rights specified in the public license below. The following considerations are for informational purposes only, are not exhaustive, and do not form part of our licenses.

Considerations for licensors: Our public licenses are intended for use by those authorized to give the public permission to use material in ways otherwise restricted by copyright and certain other rights. Our licenses are irrevocable. Licensors should read and understand the terms and conditions of the license they choose before applying it. Licensors should also secure all rights necessary before applying our licenses so that the public can reuse the material as expected. Licensors should clearly mark any material not subject to the license. This includes other CC-licensed material, or material used under an exception or limitation to copyright. More considerations for licensors.

Considerations for the public: By using one of our public licenses, a licensor grants the public permission to use the licensed material under specified terms and conditions. If the licensors permission is not necessary for any reason for example, because of any applicable exception or limitation to copyright then that use is not regulated by the license. Our licenses grant only permissions under copyright and certain other rights that a licensor has authority to grant. Use of the licensed material may still be restricted for other reasons, including because others have copyright or other rights in the material. A licensor may make special requests, such as asking that all changes be marked or described. Although not required by our licenses, you are encouraged to respect those requests where reasonable. More considerations for the public.

Creative Commons Attribution 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of

these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

## Section 1 Definitions.

a. Adapted Material means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.

b. Adapter's License means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.

c. Copyright and Similar Rights means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.

d. Effective Technological Measures means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.

e. Exceptions and Limitations means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.

f. Licensed Material means the artistic or literary work, database, or other material to which the Licensor applied this Public License.

g. Licensed Rights means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.

h. Licensor means the individual(s) or entity(ies) granting rights under this Public License.

i. Share means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.

j. Sui Generis Database Rights means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.

k. You means the individual or entity exercising the Licensed Rights under this Public License. Your has a corresponding meaning.

## Section 2 Scope.

### a. License grant.

1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:

A. reproduce and Share the Licensed Material, in whole or in part; and

B. produce, reproduce, and Share Adapted Material.

2. Exceptions and Limitations. For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.

3. Term. The term of this Public License is specified in Section 6(a).

4. Media and formats; technical modifications allowed. The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section 2(a)(4) never produces Adapted Material.

5. Downstream recipients.

A. Offer from the Licensor Licensed Material. Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.

B. No downstream restrictions. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.

6. No endorsement. Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i).

b. Other rights.

1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.

2. Patent and trademark rights are not licensed under this Public License.

3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties.

### Section 3 License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

a. Attribution.

1. If You Share the Licensed Material (including in modified form), You must:

A. retain the following if it is supplied by the Licensor with the Licensed Material:

i. identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);

ii. a copyright notice;

iii.

a notice that refers to this Public License;

iv. a notice that refers to the disclaimer of warranties;

v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;

B. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and

C. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.

2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.

3. If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.

4. If You Share Adapted Material You produce, the Adapter's License You apply must not prevent recipients of the Adapted Material from complying with this Public License.

#### Section 4 Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

- a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database;
- b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material; and
- c. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

#### Section 5 Disclaimer of Warranties and Limitation of Liability.

- a. Unless otherwise separately undertaken by the Licensor, to the extent possible, the Licensor offers the Licensed Material as-is and as-available, and makes no representations or warranties of any kind concerning the Licensed Material, whether express, implied, statutory, or other. This includes, without limitation, warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence of latent or other defects, accuracy, or the presence or absence of errors, whether or not known or discoverable. Where disclaimers of warranties are not allowed in full or in part, this disclaimer may not apply to You.
- b. To the extent possible, in no event will the Licensor be liable to You on any legal theory (including, without limitation, negligence) or otherwise for any direct, special, indirect, incidental, consequential, punitive, exemplary, or other losses, costs, expenses, or damages arising out of this Public License or use of the Licensed Material, even if the Licensor has been advised of the possibility of such losses, costs, expenses, or damages. Where a limitation of liability is not allowed in full or in part, this limitation may not apply to You.
- c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

#### Section 6 Term and Termination.

- a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.
- b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:
  1. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or
  2. upon express reinstatement by the Licensor.

c. For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

d. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.

e. Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

## Section 7 Other Terms and Conditions.

a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.

b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

## Section

### 8 Interpretation.

a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully be made without permission under this Public License.

b. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions.

c. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.

d. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.

Creative Commons is not a party to its public licenses. Notwithstanding, Creative Commons may elect to apply one of its public licenses to material it publishes and in those instances will be considered the Licensor. Except for the limited purpose of indicating that material is shared under a Creative Commons public license or as otherwise permitted by the Creative Commons policies published at [creativecommons.org/policies](https://creativecommons.org/policies), Creative Commons does not authorize the use of the trademark Creative Commons or any other trademark or logo of Creative Commons without its prior written consent including, without limitation, in connection with any unauthorized modifications to any of its public licenses or any other arrangements, understandings, or agreements concerning use of licensed material. For the avoidance of doubt, this paragraph does not form part of the public licenses.

Creative Commons may be contacted at [creativecommons.org](https://creativecommons.org).

Copyright (c) <year> <owner>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;

OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copying and distribution of this file, with or without modification, are permitted in any medium without royalty provided the copyright notice and this notice are preserved. This file is offered as-is, without any warranty.

# Userspace RCU library licensing

This project is compliant with the [REUSE](<https://reuse.software/spec/>) specification, each file contains [SPDX](<https://spdx.org/specifications>) tags, the full licenses can be found in the `LICENSES/` directory.

## Library code

The library part is distributed under `LGPL-2.1-or-later`. See `LICENSES/LGPL-2.1-or-later.txt` for license details. Refer to the individual file headers for details.

LGPL-compatible source code can statically use the library header using :

```
****
#define _LGPL_SOURCE
#include <urcu.h>
****
```

Dynamic-only linking with the LGPL library is used if `\_LGPL\_SOURCE` is not defined. It permits relinking with newer versions of the library, which is required by the LGPL license.

See `lgpl-relicensing.md` for details.



## ## Headers

Most headers are also distributed under `LGPL-2.1-or-later`, some include code from `atomic\_ops` and are distributed under `Boehm-GC` and some others are distributed under `MIT`. Refer to the individual file headers for details.

xchg() primitive has been rewritten from scratch starting from `atomic\_ops 1.2` which has a MIT-style license that is intended to allow use in both free and proprietary software:

\*\*\*\*

[http://www.hpl.hp.com/research/linux/atomic\\_ops/LICENSING.txt](http://www.hpl.hp.com/research/linux/atomic_ops/LICENSING.txt)

[http://www.hpl.hp.com/personal/Hans\\_Boehm/gc/gc\\_source/](http://www.hpl.hp.com/personal/Hans_Boehm/gc/gc_source/)

\*\*\*\*

## ## Tests

Library test code is distributed under the `GPL-2.0-only` license. See `LICENSES/GPL-2.0-only.txt` for license details. See headers of individual files under `tests/` for details.

## ## Documentation

The documentation is distributed under the `CC-BY-4.0` license.

## ## Build system

Build system related files are mostly distributed under the `MIT` license. See headers of individual files for details.

## ## Generated data files

Generated data files are distributed under the `CC-1.0` license.  
Creative Commons Legal Code

CC0 1.0 Universal

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS DOCUMENT DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS

PROVIDED HEREUNDER, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED HEREUNDER.

## Statement of Purpose

The laws of most jurisdictions throughout the world automatically confer exclusive Copyright and Related Rights (defined below) upon the creator and subsequent owner(s) (each and all, an "owner") of an original work of authorship and/or a database (each, a "Work").

Certain owners wish to permanently relinquish those rights to a Work for the purpose of contributing to a commons of creative, cultural and scientific works ("Commons")

that the public can reliably and without fear of later claims of infringement build upon, modify, incorporate in other works, reuse and redistribute as freely as possible in any form whatsoever and for any purposes, including without limitation commercial purposes. These owners may contribute to the Commons to promote the ideal of a free culture and the further production of creative, cultural and scientific works, or to gain reputation or greater distribution for their Work in part through the use and efforts of others.

For these and/or other purposes and motivations, and without any expectation of additional consideration or compensation, the person associating CC0 with a Work (the "Affirmer"), to the extent that he or she is an owner of Copyright and Related Rights in the Work, voluntarily elects to apply CC0 to the Work and publicly distribute the Work under its terms, with knowledge of his or her Copyright and Related Rights in the Work and the meaning and intended legal effect of CC0 on those rights.

1. Copyright and Related Rights. A Work made available under CC0 may be protected by copyright and related or neighboring rights ("Copyright and Related Rights"). Copyright and Related Rights include, but are not limited to, the following:

- i. the right to reproduce, adapt, distribute, perform, display, communicate, and translate a Work;
- ii. moral rights retained by the original author(s) and/or performer(s);
- iii. publicity and privacy rights pertaining to a person's image or likeness depicted in a Work;
- iv. rights protecting against unfair competition in regards to a Work, subject to the limitations in paragraph 4(a), below;
- v. rights protecting the extraction, dissemination, use and reuse of data in a Work;
- vi. database rights (such as those arising under Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal

protection of databases, and under any national implementation

thereof, including any amended or successor version of such directive); and

vii. other similar, equivalent or corresponding rights throughout the world based on applicable law or treaty, and any national implementations thereof.

2. Waiver. To the greatest extent permitted by, but not in contravention of, applicable law, Affirmer hereby overtly, fully, permanently, irrevocably and unconditionally waives, abandons, and surrenders all of Affirmer's Copyright and Related Rights and associated claims and causes of action, whether now known or unknown (including existing as well as future claims and causes of action), in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "Waiver"). Affirmer makes the Waiver for the benefit of each member of the public at large and to the detriment of Affirmer's heirs and successors, fully intending that such Waiver shall not be subject to revocation, rescission, cancellation, termination, or any other legal or equitable action to disrupt the quiet enjoyment of the Work by the public as contemplated by Affirmer's express Statement of Purpose.

3. Public License Fallback. Should any part of the Waiver for any reason be judged legally invalid or ineffective under applicable law, then the Waiver shall be preserved to the maximum extent permitted taking into account Affirmer's express Statement of Purpose. In addition, to the extent the Waiver is so judged Affirmer hereby grants to each affected person a royalty-free, non transferable, non sublicensable, non exclusive, irrevocable and unconditional license to exercise Affirmer's Copyright and Related Rights in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "License"). The License shall be deemed effective as of the date CC0 was applied by Affirmer to the Work. Should any part of the License for any reason be judged legally invalid or ineffective under applicable law, such partial invalidity or ineffectiveness shall not invalidate the remainder of the License, and in such case Affirmer hereby affirms that he or she will not (i) exercise any of his or her remaining Copyright and Related Rights in the Work or (ii) assert any associated claims and causes of action with respect to the Work, in either case contrary to Affirmer's express Statement of Purpose.

#### 4. Limitations and Disclaimers.

- a. No trademark or patent rights held by Affirmer are waived, abandoned, surrendered, licensed or otherwise affected by this document.
- b. Affirmer offers the Work as-is and makes no representations or warranties of any kind concerning the Work, express, implied, statutory or otherwise, including without limitation warranties of title, merchantability, fitness for a particular purpose, non infringement, or the absence of latent or other defects, accuracy, or the present or absence of errors, whether or not discoverable, all to the greatest extent permissible under applicable law.
- c. Affirmer disclaims responsibility for clearing rights of other persons that may apply to the Work or any use thereof, including without limitation any person's Copyright and Related Rights in the Work. Further, Affirmer disclaims responsibility for obtaining any necessary consents, permissions or other rights required for any use of the Work.
- d. Affirmer understands and acknowledges that Creative Commons is not a party to this document and has no duty or obligation with respect to this CC0 or use of the Work.

## 1.118 libunistring 1.3-r0

### 1.118.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that

patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS

### 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

## 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The

"System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

## 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated

conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

### 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

### 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all



recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

#### 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

#### 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the

machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years

and valid for as

long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in

the same way through the same place at no

further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source

as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in

source code form), and must require no special password or key for unpacking, reading or copying.

## 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on

those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

## 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under

this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

#### 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

#### 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

#### 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within

the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

#### 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

#### 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

#### 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.



Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

#### 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

#### 16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### 17.

Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>  
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

GNU LESSER GENERAL PUBLIC LICENSE  
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>  
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

#### 1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

#### 2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

- a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or
- b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

#### 3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

- a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the object code with a copy of the GNU GPL and this license document.

#### 4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

- a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the Combined Work with a copy of the GNU GPL and this license document.
- c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.
- d) Do one of the following:
  - 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.
  - 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.
- e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application

Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

## 5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
- b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

## 6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

# 1.119 gobject-introspection 1.84.0-r0

## 1.119.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.



2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based

on it,  
under Section 2) in object code or executable form under the terms of  
Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.

EXCEPT WHEN

OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING

WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN  
IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License,  
or  
(at your option) any later version.

This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.

The hypothetical commands `show  
w' and `show c' should show the appropriate  
parts of the General Public License. Of course, the commands you use may  
be called something other than `show w' and `show c'; they could even be  
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your  
school, if any, to sign a "copyright disclaimer" for the program, if  
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into  
proprietary programs. If your program is a subroutine library, you may  
consider it more useful to permit linking proprietary applications with the  
library. If this is what you want to do, use the GNU Lesser General  
Public License instead of this License.

GNU LIBRARY GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is  
numbered 2 because it goes with version 2 of the ordinary GPL.]

#### Preamble

The licenses for most software are designed to take away your  
freedom to share and change it. By contrast, the GNU General Public  
Licenses are intended to guarantee your freedom to share and change  
free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some  
specially designated Free Software Foundation software, and to any  
other libraries whose authors decide to use it. You can use it for  
your libraries, too.

When we speak of free software, we are referring to freedom,  
not  
price. Our General Public Licenses are designed to make sure that you  
have the freedom to distribute copies of free software (and charge for  
this service if you wish), that you receive source code or can get it  
if you want it, that you can change the software or use pieces of it  
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid  
anyone to deny you these rights or to ask you to surrender the rights.  
These restrictions translate to certain responsibilities for you if  
you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis  
or for a fee, you must give the recipients all the rights that we gave  
you. You must make sure that they, too, receive or can get the source  
code. If you link a program with the library, you must provide  
complete object files to the recipients so that they can relink them  
with the library, after making  
changes to the library and recompiling  
it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright  
the library, and (2) offer you this license which gives you legal  
permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain  
that everyone understands that there is no warranty for this free  
library. If the library is modified by someone else and passed on, we  
want its recipients to know that what they have is not the original  
version, so that any problems introduced by others will not reflect on  
the original authors' reputations.

Finally, any free program is threatened constantly by software  
patents. We wish to avoid the danger that companies distributing free  
software will individually obtain patent licenses, thus in effect  
transforming the program into proprietary software. To prevent this,  
we have made it clear that any patent must be licensed  
for everyone's  
free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary  
GNU General Public License, which was designed for utility programs. This  
license, the GNU Library General Public License, applies to certain  
designated libraries. This license is quite different from the ordinary  
one; be sure to read it in full, and don't assume that anything in it is  
the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because  
of this blurred distinction, using the ordinary General  
Public License for libraries did not effectively promote software  
sharing, because most developers did not use the libraries. We  
concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the  
users of those programs of all benefit from the free status of the  
libraries themselves. This Library General Public License is intended to  
permit developers of non-free programs to use free libraries, while  
preserving your freedom as a user of such programs to change the free  
libraries that are incorporated in them. (We have not seen how to achieve  
this as regards changes in header files, but we have achieved it as regards  
changes in the actual functions of the Library.) The hope is that this  
will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and  
modification follow. Pay close attention to the difference  
between a  
"work based on the library" and a "work that uses the library". The  
former contains code derived from the library, while the latter only  
works together with the library.

Note that it is possible for a library to be covered by the ordinary  
General Public License rather than by this special one.

## GNU LIBRARY GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which  
contains a notice placed by the copyright holder or other authorized  
party saying it may be distributed under the terms of this Library  
General Public License (also called "this License"). Each licensee is  
addressed as "you".

A "library" means a collection of software functions and/or data  
prepared so as to be conveniently linked with application programs  
(which use some of those functions and data) to form executables.



The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no

charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition,  
mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the

Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
- b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for

copying, distributing or modifying  
the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute  
the Library at all. For example, if a patent  
license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system;  
it is up to the author/donor to decide if he or she is willing  
to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries,

so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each

version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR

LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year>  
<name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.



<signature of Ty Coon>, 1 April 1990

Ty Coon, President of Vice

That's all there is to it!

gobject-introspection has two licenses; one for the typelib library,  
and one for the tools.

\* The typelib libraries (gobject-introspection/) are licensed under the LGPLv2+.  
See the file COPYING.LGPL.

\* The remaining code is GPLv2+ compatible (see the file COPYING.GPL) and  
consists of a mix of GPLv2+, LGPLv2+ and MIT. See the license headers in  
each file for details.

In general where applicable files should have headers denoting their license  
status; if they do not, please file a bug at  
<https://gitlab.gnome.org/GNOME/gobject-introspection/issues>.

## 1.120 judy 1.0.5-r1

### 1.120.1 Available under license :

Judy - C library functions for creating and accessing dynamic arrays  
Copyright (C) 2004 Doug Baskins

This library is free software; you can redistribute it and/or modify  
it under the terms of the GNU Lesser General Public License as  
published by the Free Software Foundation; either version 2.1 of the  
License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but  
WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU  
Lesser General Public License for more details.

#### GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts  
as the successor of  
the GNU Library Public License, version 2, hence  
the version number 2.1.]

## Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary

GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are

the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

## GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based

on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus,

it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once

this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the

Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library.

The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding

machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable

containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.



7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not

excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a

license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER

EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is

safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there  
is to it!

# 1.121 libnl3 3.11.0-r0

## 1.121.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE  
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

## Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is

modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of

free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

## GNU LESSER GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from

such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If



identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

### 3. You may opt to apply the terms of the ordinary GNU General Public

License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the

source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the

copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany

the work with the complete corresponding

machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate

properly with a modified version of the library, if

the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally

distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that

everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey

the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990

Ty Coon, President of Vice

That's all there is to it!

## 1.122 zstd 1.5.7-r0

## 1.122.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we



want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1

above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable

source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the

Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in

certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY

YOU OR

THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU

General Public License as published by  
the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along  
with this program; if not, write to the Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this  
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute

it

under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate

parts of the General Public License. Of course, the commands you use may be called something other than `show w` and `show c`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.  
BSD License

For Zstandard software

Copyright (c) Meta Platforms, Inc. and affiliates. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name Facebook, nor Meta, nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON

ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# 1.123 vim 9.1.1566-r0

## 1.123.1 Available under license :

\*uganda.txt\* For Vim version 9.1. Last change: 2025 Jul 08

VIM REFERENCE MANUAL by Bram Moolenaar

\*uganda\* \*Uganda\* \*copying\* \*copyright\* \*license\*

### SUMMARY

\*iccf\* \*ICCF\*

Vim is Charityware. You can use and copy it as much as you like, but you are encouraged to make a donation for needy children in Uganda. Please see |kcc| below or visit the ICCF web site, available at these URLs:

<https://iccf-holland.org/>

<https://www.vim.org/iccf/>

<https://www.iccf.nl/>

You can also sponsor the development of Vim, see |sponsor|. The money goes to Uganda anyway.

The Open Publication License applies to the Vim documentation, see |manual-copyright|.

The icons used for the toolbar in the Win32 GUI are from the "Farm-Fresh Web Icons" set. These icons are licensed under the Creative Commons Attribution 3.0 License and were created by FatCow Web Hosting. They were previously available at <http://www.fatcow.com/free-icons> and are still available at: <https://www.iconarchive.com/show/farm-fresh-icons-by-fatcow.1.html>

===

begin of license ===

### VIM LICENSE

- I) There are no restrictions on distributing unmodified copies of Vim except that they must include this license text. You can also distribute unmodified parts of Vim, likewise unrestricted except that they must include this license text. You are also allowed to include executables that you made from the unmodified Vim sources, plus your own usage



examples and Vim scripts.

II) It is allowed to distribute a modified (or extended) version of Vim, including executables and/or source code, when the following four conditions are met:

- 1) This license text must be included unmodified.
- 2) The modified Vim must be distributed in one of the following five ways:

- a) If you make changes to Vim yourself, you must clearly describe in the distribution how to contact you. When the maintainer asks you (in any way) for a copy of the modified Vim you distributed, you must

make your changes, including source code, available to the maintainer without fee. The maintainer reserves the right to include your changes in the official version of Vim. What the maintainer will do with your changes and under what license they will be distributed is negotiable. If there has been no negotiation then this license, or a later version, also applies to your changes. The current maintainers are listed here: <https://github.com/orgs/vim/people>. If this changes it will be announced in appropriate places (most likely [vim.sf.net](http://vim.sf.net), [www.vim.org](http://www.vim.org) and/or [comp.editors](http://comp.editors)). When it is completely impossible to contact the maintainer, the obligation to send him your changes ceases. Once the maintainer has confirmed that he has received your changes they will not have to be sent again.

- b) If you have received a modified Vim that was distributed as mentioned under a) you are allowed to further distribute it unmodified, as mentioned at I).

If you make additional changes the text under a) applies to those changes.

- c) Provide all the changes, including source code, with every copy of the modified Vim you distribute. This may be done in the form of a context diff. You can choose what license to use for new code you add. The changes and their license must not restrict others from making their own changes to the official version of Vim.

- d) When you have a modified Vim which includes changes as mentioned under c), you can distribute it without the source code for the changes if the following three conditions are met:

- The license that applies to the changes permits you to distribute the changes to the Vim maintainer without fee or restriction, and permits the Vim maintainer to include the changes in the official version of Vim without fee or restriction.
- You keep the changes for at least three years after last distributing the corresponding modified Vim. When

the maintainer

or someone who you distributed the modified Vim to asks you (in any way) for the changes within this period, you must make them available to him.

- You clearly describe in the distribution how to contact you. This contact information must remain valid for at least three years

after last distributing the corresponding modified Vim, or as long as possible.

e) When the GNU General Public License (GPL) applies to the changes, you can distribute the modified Vim under the GNU GPL version 2 or any later version.

3) A message must be added, at least in the output of the ":version" command and in the intro screen, such that the user of the modified Vim is able to see that it was modified. When distributing as mentioned under 2)e) adding the message is only required for as far as this does not conflict with the license used for the changes.

4) The contact information as required under 2)a) and 2)d) must not be removed or changed, except that the person himself can make corrections.

III) If you distribute a modified version of Vim, you are encouraged to use the Vim license for your changes and make them available to the maintainer, including the source code. The preferred way to do this is by e-mail or by uploading the files to a server and e-mailing the URL. If the number of changes is small (e.g., a modified Makefile) e-mailing a context diff will do. The e-mail address to be used is <maintainer@vim.org>

IV) It is not allowed to remove this license from the distribution of the Vim sources, parts of it or from a modified version. You may use this license for previous Vim releases instead of the license that they came with, at your option.

=== end of license ===

Note:

- If you are happy with Vim, please express that by reading the rest of this file and consider helping needy children in Uganda.
- If you want to support further Vim development consider becoming a [sponsor]. The money goes to Uganda anyway.
- According to Richard Stallman the Vim license is GNU GPL compatible. A few minor changes have been made since he checked it, but that should not make a difference.
- If you link Vim with a library that goes under the GNU GPL, this limits further distribution to the GNU GPL. Also when you didn't actually change anything in Vim.
- Once a change is included that goes under the GNU GPL, this forces all

further changes to also be made under the GNU GPL or a compatible license.

- If you distribute a modified version of Vim, you can include your name and contact information with the "--with-modified-by" configure argument or the MODIFIED\_BY define.

- This license applies to the source code of Vim. However, different license conditions may apply to some runtime files included with Vim; these will be specified in the header of each respective file.

=====

Kibaale

Children's Centre \*kcc\* \*Kibaale\* \*charity\*

Kibaale Children's Centre (KCC) is located in Kibaale, a small town in the south of Uganda, near Tanzania, in East Africa. The area is known as Rakai District. The population is mostly farmers. Although people are poor, there usually is enough food. But this district is suffering from AIDS more than any other part of the world. Some say that it started there. Estimations are that in the past 10 to 30% of the Ugandans are infected with HIV. Because parents die, there are many orphans. In this district about 60,000 children have lost one or both parents, out of a population of 350,000. Although AIDS is now mostly under control, the problems are still continuing.

The children need a lot of help. The KCC is working hard to provide the needy with food, medical care and education. Food and medical care to keep them healthy now, and education so that they can take care of themselves in the future. KCC works on a Christian base, but help is given to children of any religion.

The key to solving the problems in this area is education. This has been neglected in the past years with president Idi Amin and the following civil wars. Now that the government is stable again, the children and parents have to learn how to take care of themselves and how to avoid infections. There is also help for people who are ill and hungry, but the primary goal is to prevent people from getting ill and to teach them how to grow healthy food.

Most of the orphans are living in an extended family. An uncle or older sister is taking care of them. Because these families are big and the income (if any) is low, a child is lucky if it gets healthy food. Clothes, medical care and schooling is beyond its reach. To help these needy children, a sponsorship program was put into place. A child can be financially adopted. For a few dollars a month KCC sees to it that the child gets indispensable items, is healthy, goes to school and KCC takes care of anything else that needs to be done for the child and the family that supports it.

Besides helping the child directly, the environment where the child grows up

needs to be improved. KCC helps schools to improve their teaching methods. There is a demonstration school at the centre and teacher trainings are given. Health workers are being trained, hygiene education is carried out and households are stimulated to build a proper latrine. I helped setting up a production site for cement slabs. These are used to build a good latrine. They are sold below cost price.

There is a clinic at the project, which provides children and their family medical help. Since 2020 a maternity ward was added and 24/7 service is available. When needed, transport to a hospital is offered. Immunization programs are carried out and help is provided when an epidemic is breaking out (measles and cholera have been a problem).

\*donate\*

Summer

1994 to summer 1995 I spent a whole year at the centre, working as a volunteer. I have helped to expand the centre and worked in the area of water and sanitation. I learned that the help that the KCC provides really helps. When I came back to Holland, I wanted to continue supporting KCC. To do this I'm raising funds and organizing the sponsorship program. Please consider one of these possibilities:

1. Sponsor a child in primary school: 17 euro a month (or more).
2. Sponsor a child in secondary school: 25 euro a month (or more).
3. Sponsor the clinic: Any amount a month or quarter
4. A one-time donation

Compared with other organizations that do child sponsorship the amounts are very low. This is because the money goes directly to the centre. Less than 5% is used for administration. This is possible because this is a small organization that works with volunteers. If you would like to sponsor a child, you should have the intention to do this for at least one year.

How do you know that the money will be spent right? First of all you have my personal guarantee as the author of Vim. I trust the people that are working at the centre, I know them personally. Furthermore, the centre has been co-sponsored and inspected by World Vision, Save the Children Fund and is now under the supervision of Pacific Academy Outreach Society. The centre is visited about once a year to check the progress (at our own cost). I have visited the centre myself many times, starting in 1993. The visit reports are on the ICCF web site.

If you have any further questions, send e-mail: <Bram@vim.org>.

The address of the centre is:

Kibaale Children's Centre  
p.o. box 1658  
Masaka, Uganda, East Africa

Sending money:    \*iccf-donations\*

Check the ICCF web site for the latest information! See |iccf| for the URL.

USA: The methods mentioned below can be used.

If you must send a check send it to our Canadian partner:

<https://www.kuwasha.net/>

Canada: Contact

Kuwasha in Surrey, Canada. They take care of the Canadian sponsors for the children in Kibaale. Kuwasha forwards 100% of the money to the project in Uganda. You can send them a one time donation directly.

Look on their site for information about sponsorship:

<https://www.kuwasha.net/>

If you make a donation to Kuwasha you will receive a tax receipt which can be submitted with your tax return.

Holland: Transfer to the account of "Stichting ICCF Holland" in Amersfoort. This will allow for tax deduction if you live in Holland. ING bank, IBAN: NL95 INGB 0004 5487 74

Germany: It is possible to make donations that allow for a tax return.

Check the ICCF web site for the latest information:

<https://iccf-holland.org/germany.html>

Europe: Use a bank transfer if possible. See "Others" below for the swift code and IBAN number.

Any other method should work. Ask for information about sponsorship.

Credit Card: You can use PayPal to send money with a Credit card. This is

the most widely used Internet based payment system. It's really simple to use. Use this link to find more info:

[https://www.paypal.com/en\\_US/mrb/pal=XAC62PML3GF8Q](https://www.paypal.com/en_US/mrb/pal=XAC62PML3GF8Q)

The e-mail address for sending the money to is:

[Bram@iccf-holland.org](mailto:Bram@iccf-holland.org)

Others: Transfer to this account if possible:

ING bank: IBAN: NL95 INGB 0004 5487 74

Swift code: INGBNL2A

under the name "stichting ICCF Holland", Amersfoort

Checks are not accepted.

vim:tw=78:ts=8:noet:ft=help:norl:

Unless otherwise stated, all files are distributed under the Simplified BSD license included below.

Copyright (c) 2011-2019 EditorConfig Team  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENZA VIM

Nota: Agli effetti legali ha valore non la seguente traduzione italiana, ma l'originale inglese su cui basata. alla versione inglese che si fa riferimento qui sotto quando si parla di "questa licenza" (NdT).

- I) Non ci sono restrizioni alla distribuzione di copie non modificate di Vim, alla condizione che esse includano il testo di questa licenza. Si possono anche distribuire parti non modificate di Vim, sempre senza restrizioni, alla condizione che esse includano il testo di questa licenza. anche consentito includere programmi eseguibili da voi preparati a partire dai file sorgenti Vim non modificati, pi i vostri esempi di utilizzo e i vostri script Vim.
- II) consentito distribuire una versione modificata (o estesa) di Vim, contenente programmi eseguibile e/o codice sorgente, quando siano rispettare le seguenti quattro condizioni:
- 1) Il testo di questa licenza deve essere incluso senza modifiche.
  - 2) La versione modificata di Vim

dev'essere distribuita in una della

seguenti cinque maniere:

- a) Se voi personalmente modificate a Vim, dovete descrivere chiaramente nella distribuzione come possibile mettersi in contatto con voi. Quando il manutentore vi chiede (in qualsiasi forma) una copia della versione modificata di Vim da voi distribuita, dovete rendere disponibili le vostre modifiche, compreso il codice sorgente, al manutentore, senza alcuna spesa. Il manutentore si riserva il diritto di includere le vostre modifiche nella versione ufficiale di Vim. Quel che il manutentore potrà fare con le vostre modifiche e sotto quale licenza esse verranno distribuite può essere oggetto di negoziazione. Se non c'è stata alcuna negoziazione, questa licenza, o una versione successiva della stessa, anche applicabile alle vostre modifiche.

I manutentori correnti sono elencati qui:

<https://github.com/orgs/vim/people>.

Se dovessero cambiare, ne sarà dato annuncio nei luoghi appropriati (verosimilmente [vim.sf.net](http://vim.sf.net), [www.vim.org](http://www.vim.org) e/o [comp.editors](mailto:comp.editors)). Quando sia completamente impossibile contattare il manutentore, l'obbligo di spedirgli le vostre modifiche viene meno. Una volta che il manutentore abbia confermato di aver ricevuto le vostre modifiche, non sarà necessario spedirle ulteriormente.

- b) Se avete ricevuto un Vim modificato che era stato distribuito come descritto al punto a) potete ulteriormente distribuirlo senza modificarlo, come menzionato al punto I). Se fate ulteriori modifiche, il testo al punto a) si applica a queste modifiche.
- c) Includete tutte le modifiche, compreso il codice sorgente, con ogni copia del Vim modificato da voi distribuito. Questo può essere fatto sotto forma dell'output di un comando "diff", che contenga anche

il contesto [ossia anche le linee non modificate di sorgente

che vengono prima o dopo le modifiche - NdT]. Potete scegliere quale licenza volete utilizzare per il nuovo codice sorgente che voi aggiungere. Le modifiche e la relativa licenza non devono impedire a terze parti di apportare le loro proprie modifiche a Vim.

- d) Quando avete un Vim modificato che comprende modifiche quali quelle menzionate al punto c), potete distribuirlo senza il codice sorgente che descrive le modifiche, se le seguenti tre condizioni sono soddisfatte:

- La licenza applicabile alle modifiche vi consente di distribuire le modifiche al manutentore Vim senza alcun costo o restrizione, e permette al manutentore Vim di includere le modifiche nella versione ufficiale di Vim senza alcun costo o restrizione.
- Le modifiche da voi fatte sono conservate per almeno tre anni,

dopo che avrete distribuito per l'ultima volta il Vim modificato corrispondente alle modifiche stesse. Quando il manutentore o qualcuno a cui voi avete distribuito il Vim modificato vi chiede

(in qualsiasi forma) le modifiche, entro il periodo qui indicato, dovete fargliele avere.

- Dovete descrivere chiaramente nella distribuzione come possibile mettersi in contatto con voi. Questa informazione di contatto deve rimanere valida per almeno tre anni dopo l'ultima data di distribuzione della corrispondente modifica di Vim, oppure il più a lungo possibile.

e) Quando al GNU General Public License (GPL) si applica alle modifiche, potete distribuire il Vim modificato sotto la versione 2 della licenza GNU GPL versione 2 o una qualsiasi versione successiva della stessa licenza.

3) Un messaggio deve essere aggiunto, almeno all'output del comando ":version" e nella schermata introduttiva, in modo che l'utilizzatore del Vim modificato sia in grado di comprendere cosa è stato modificato. Quando si distribuisca come menzionato al punto 2)e) l'aggiunta del messaggio richiesta solo se ci non conduca a conflitti con la licenza usata per le modifiche.

4) L'informazione di contatto, richiesta ai punti 2)a) e 2)d) non deve essere rimossa o modificata, tranne che per il fatto che la persona stessa può apportare delle correzioni.

III) Se distribuite una versione modificata di Vim, siete incoraggiati a utilizzare la licenza Vim per le vostre modifiche, e a mettere le modifiche a disposizione del manutentore, compreso il codice sorgente. Il modo preferito in cui fare questo è tramite e-mail o caricando i file su un server ed inviandone l'indirizzo (URL) tramite e-mail. Se il numero di modifiche è piccolo (per esempio, una modifica a un Makefile) inviare tramite

email l'output del comando diff con contesto

può essere sufficiente. L'indirizzo e-mail da utilizzare

<maintainer@vim.org>

IV) Non è consentito rimuovere questa licenza dalla distribuzione dei codici sorgente Vim, anche in parte, o da una versione di Vim modificata. Potete usare questa licenza per precedenti versioni di Vim, invece della licenza con cui esse sono state distribuite, a vostra scelta.

#### LICENZA DO VIM

I) Não há restrições para distribuir cópias não modificadas do Vim, exceto que elas devem incluir este texto de licença. Você também pode distribuir partes não modificadas do Vim, igualmente irrestritas, exceto que elas devem incluir este texto de licença. Você também tem permissão para incluir executáveis que você fez a partir das fontes não modificadas do Vim, além de seus próprios exemplos de uso e scripts do Vim.

II) permitido distribuir uma versão modificada (ou estendida) do Vim, incluindo executáveis e/ou código-fonte, quando as quatro condições a seguir forem atendidas:



- 1) Este texto de licena deve ser includo sem modificaes.
- 2) O Vim modificado deve ser distribudo de uma das cinco formas a seguir:
  - a) Se voc fizer alteraes no Vim, dever descrever claramente na distribuio como entrar em contato com voc. Quando o mantenedor solicitar a voc (de qualquer forma) uma cpia do Vim modificado que voc distribuiu, voc dever disponibilizar suas alteraes, incluindo o cdigo-fonte, ao mantenedor sem taxa. O mantenedor reserva o direito de incluir suas alteraes na verso oficial do Vim. O que o mantenedor far com suas alteraes e sob qual licena elas sero distribudas negociavel. Se no houve negcio ento esta licena, ou uma verso posterior, tambm se aplica s suas alteraes. Os mantenedores atuais esto listados aqui: <https://github.com/orgs/vim/people>. Se isso mudar, ser anunciado em lugares apropriados (provavelmente [vim.sf.net](http://vim.sf.net), [www.vim.org](http://www.vim.org) e/ou [comp.editors](mailto:comp.editors)). Quando for completamente impossvel contatar o mantenedor, a obrigao de enviar a ele suas alteraes cessa. Uma vez que o mantenedor tenha confirmado que ele recebeu suas alteraes, elas no precisaro ser enviadas novamente.
  - b) Se voc recebeu um Vim modificado que foi distribudo como mencionado em a), voc tem permissso para distribu-lo sem modificaes, como mencionado em 1). Se voc fizer alteraes adicionais, o texto em a) se aplica a essas alteraes.
  - c) Fornea todas as alteraes, incluindo o cdigo-fonte, com cada cpia do Vim modificado que voc distribuir. Isso pode ser feito na forma de um diff de contexto. Voc pode escolher qual licena usar para o novo cdigo que voc adicionar. As alteraes e sua licena no devem restringir outros de fazer suas prprias alteraes na verso oficial do Vim.
  - d) Quando voc tem um Vim modificado que inclui alteraes conforme mencionado em c), voc pode distribu-lo sem o cdigo-fonte para as alteraes se as trs condies a seguir forem atendidas:
    - A licena que se aplica s alteraes permite que voc distribua as alteraes para o mantenedor do Vim sem taxa ou restrio, e permite que o mantenedor do Vim inclua as alteraes na verso oficial do Vim sem taxa ou restrio.
    - Voc mantm as alteraes por pelo menos trs anos aps a ltima distribuio do Vim modificado correspondente. Quando o mantenedor ou algum para quem voc distribuiu o Vim modificado solicitar a voc (de qualquer forma) as alteraes dentro deste perodo, voc deve disponibiliz-las a ele.
    - Voc descreve claramente na distribuio como entrar em contato com voc. Estas informaes de contato devem permanecer vlidas por pelo menos trs anos aps a ltima distribuio do Vim modificado correspondente, ou pelo maior tempo possvel.
  - e) Quando a Licena Pblica Geral GNU (GPL) se aplica s alteraes,

você pode distribuir o Vim modificado sob a GNU GPL versão 2 ou qualquer versão posterior.

- 3) Uma mensagem deve ser adicionada, no mínimo na saída do comando `":version"` e na tela de introdução, de modo que o usuário do Vim modificado seja capaz de ver que ele foi modificado.

Ao distribuir

conforme mencionado em 2)e), é necessário adicionar a mensagem na medida em que isso não entre em conflito com a licença usada para as alterações.

- 4) As informações de contato conforme exigido em 2)a) e 2)d) não devem ser removidas ou alteradas, exceto que a própria pessoa pode fazer correções.

III) Se você distribuir uma versão modificada do Vim, encorajamos que use a licença do Vim para suas alterações e disponibilizá-las ao mantenedor, incluindo o código-fonte. A maneira preferida de fazer isso por e-mail ou enviando os arquivos para um servidor e enviando a URL por e-mail. Se o número de alterações for pequeno (por exemplo, um Makefile modificado), enviar um diff de contexto por e-mail será suficiente.

O endereço de e-mail a ser usado <maintainer@vim.org>

IV) Não é permitido remover esta licença da distribuição das fontes do Vim, partes dela ou de uma versão modificada. Você pode usar esta licença para versões anteriores do Vim em vez da licença com a qual elas vieram, a seu critério.

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You

can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser

General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a

fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves,

then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License.

Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)



- b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under

the terms of the  
Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any

particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of

all derivatives of our free software and of promoting the sharing and reuse of software generally.

## NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public

License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!

Some code in editorconfig-vim is derived from code licensed under the PSF license. The following is the text of that license, retrieved 2019-05-05 from <https://docs.python.org/2.6/license.html#terms-and-conditions-for-accessing-or-otherwise-using-python>

#### PSF LICENSE AGREEMENT FOR PYTHON 2.6.9

1. This LICENSE AGREEMENT is between the Python Software Foundation (`PSF'), and the Individual or Organization (`Licensee') accessing and otherwise using Python 2.6.9 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 2.6.9 alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., ``Copyright (c) 2001-2010 Python Software Foundation; All Rights Reserved" are retained in Python 2.6.9 alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 2.6.9 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 2.6.9.

4. PSF is making Python 2.6.9 available to Licensee on an ``AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 2.6.9 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 2.6.9 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 2.6.9, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python 2.6.9, Licensee agrees to be bound by the terms and conditions of this License Agreement.

# vi: set ft=:

VIM LICENSE

I) There are no restrictions on distributing unmodified copies of Vim except that they must include this license text. You can also distribute unmodified parts of Vim, likewise unrestricted except that they must include this license text. You are also allowed to include executables that you made from the unmodified Vim sources, plus your own usage examples and Vim scripts.

II) It is allowed to distribute a modified (or extended) version of Vim, including executables and/or source code, when the following four conditions are met:

- 1) This license text must be included unmodified.
- 2) The modified Vim must be distributed in one of the following five ways:
  - a) If you make changes to Vim yourself, you must clearly describe in

the distribution how to contact you. When the maintainer asks you (in any way) for a copy of the modified Vim you distributed, you must make your changes, including source code, available to the

maintainer without fee. The maintainer reserves the right to include your changes in the official version of Vim. What the maintainer will do with your changes and under what license they will be distributed is negotiable. If there has been no negotiation then this license, or a later version, also applies to your changes. The current maintainers are listed here: <https://github.com/orgs/vim/people>. If this changes it will be announced in appropriate places (most likely vim.sf.net, www.vim.org and/or comp.editors). When it is completely impossible to contact the maintainer, the obligation to send him your changes ceases. Once the maintainer has confirmed that he has received your changes they will not have to be sent again.

- b) If you have received a modified Vim that was distributed as mentioned under a) you are allowed to further distribute it unmodified, as mentioned at I). If you make additional changes the text under a) applies to those changes.
  - c) Provide all the changes, including source code, with every copy of the modified Vim you distribute. This may be done in the form of a context diff. You can choose what license to use for new code you add. The changes and their license must not restrict others from making their own changes to the official version of Vim.
  - d) When you have a modified Vim which includes changes as mentioned under c), you can distribute it without the source code for the changes if the following three conditions are met:
    - The license that applies to the changes permits you to distribute the changes to the Vim maintainer without fee or restriction, and permits the Vim maintainer to include the changes in the official version of Vim without fee or restriction.
    - You keep the changes for at least three years after last distributing the corresponding modified Vim. When the maintainer or someone who you distributed the modified Vim to asks you (in any way) for the changes within this period, you must make them available to him.
    - You clearly describe in the distribution how to contact you. This contact information must remain valid for at least three years after last distributing the corresponding modified Vim, or as long as possible.
  - e) When the GNU General Public License (GPL) applies to the changes, you can distribute the modified Vim under the GNU GPL version 2 or any later version.
- 3) A message must be added, at least in the output of the ":version" command and in the intro screen, such that the user of the modified Vim is able to see that it was modified. When distributing as mentioned under 2)e) adding the message is only required for as far as this does not conflict with the license used for the changes.

4) The contact information as required under 2)a) and 2)d) must not be removed or changed, except that the person himself can make corrections.

III) If you distribute a modified version of Vim, you are encouraged to use the Vim license for your changes and make them available to the maintainer, including the source code. The preferred way to do this is by e-mail or by uploading the files to a server and e-mailing the URL. If the number of changes is small (e.g., a modified Makefile) e-mailing a context diff will do. The e-mail address to be used is  
<maintainer@vim.org>

IV) It is not allowed to remove this license from the distribution of the Vim sources, parts of it or from a modified version. You may use this license for previous Vim releases instead of the license that they came with,  
at your option.

/\*

\* Copyright (C) 1989-95 GROUPE BULL

\*

\* Permission is hereby granted, free of charge, to any person obtaining a copy  
\* of this software and associated documentation files (the "Software"), to  
\* deal in the Software without restriction, including without limitation the  
\* rights to use, copy, modify, merge, publish, distribute, sublicense, and/or  
\* sell copies of the Software, and to permit persons to whom the Software is  
\* furnished to do so, subject to the following conditions:

\*

\* The above copyright notice and this permission notice shall be included in  
\* all copies or substantial portions of the Software.

\*

\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
\* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
\* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL  
\* GROUPE BULL BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN  
\* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT  
OF OR IN  
\* CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

\*

\* Except as contained in this notice, the name of GROUPE BULL shall not be  
\* used in advertising or otherwise to promote the sale, use or other dealings  
\* in this Software without prior written authorization from GROUPE BULL.

\*/

Arnaud LE HORS    BULL Research FRANCE -- Koala Project  
(XPM - X PixMap format version 2 & 3)

Internet:    lehors@sophia.inria.fr

Surface Mail:    Arnaud LE HORS, INRIA - Sophia Antipolis,  
2004, route des Lucioles, 06565 Valbonne Cedex -- FRANCE



Voice phone: (33) 93.65.77.71, Fax: (33) 93 65 77 66, Telex: 97 00 50 F

The MIT License

Copyright (c) 2008 Paul Evans <leonerdd@leonerdd.org.uk>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Unless otherwise stated, all files in this directory are distributed under the Zero-Clause BSD license.

Zero-Clause BSD

=====

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted.

THE SOFTWARE IS PROVIDED AS IS AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

VIM

I) Vim

.

Vim

.

,

Vim,  
Vim.

II) ( ) Vim,

/ ,

:

1)

Vim .

2) Vim

:

a) Vim,

, .

( )

Vim, .

Vim.

, ,

,

,

. ,

.,

, , :

<https://github.com/orgs/vim/people>.

,

(

vim.sf.net, [www.vim.org](http://www.vim.org) / comp.editors).

,

.

,

Vim .

b) Vim,

, . II) . 2) . )

. I) .

,

, . II) . 2) . a).

c) Vim

,

.

.,

,

.

Vim.

d) Vim,

. II) . 2 . c),

,

:

- ,

Vim

,

Vim.

-

Vim.

Vim

( )

.

- Vim

, .

Vim

.

e) , Vim,

GNU General Public License (GNU GPL),

Vim

GNU GPL 2

.

3) ,

":version" Vim,

Vim,

Vim ,

.

, . II) . 2) . e)

,

, .

4) , . II) . 2)

. a) . II) . 2) . d)

, , .

III) Vim

Vim

,

.

URL- .

(, Makefile),

.

URL-

[maintainer@vim.org](mailto:maintainer@vim.org)

IV)

Vim, .

Vim, ,

.

## 1.124 protobuf-c 1.5.2-r0

### 1.124.1 Available under license :

Copyright (c) 2008-2025, Dave Benson and the protobuf-c authors.  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The code generated by the protoc-gen-c code generator and by the protoc-c compiler is owned by the owner of the input files used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is covered by the above license.

# 1.125 busybox 1.37.0-r19

## 1.125.1 Available under license :

bzip2 applet in busybox is based on lightly-modified source  
of bzip2 version 1.0.4. bzip2 source is distributed  
under the following conditions (copied verbatim from LICENSE file)

=====

This program, "bzip2", the associated library "libbzip2", and all  
documentation, are copyright (C) 1996-2006 Julian R Seward. All  
rights reserved.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions  
are met:

1. Redistributions of source code must retain the above copyright  
notice, this list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented; you must  
not claim that you wrote the original software. If you use this  
software in a product, an acknowledgment in the product  
documentation would be appreciated but is not required.
3. Altered source versions must be plainly marked as such, and must  
not be misrepresented  
as being the original software.
4. The name of the author may not be used to endorse or promote  
products derived from this software without specific prior written  
permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS  
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED  
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY  
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL  
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE  
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS  
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,  
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING  
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS  
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, Cambridge, UK.  
jseward@bzip.org

bzip2/libbzip2 version 1.0.4 of 20 December 2006

--- A note on GPL versions

BusyBox is distributed under version 2 of the General Public License (included in its entirety, below). Version 2 is the only version of this license which this version of BusyBox (or modified versions derived from this one) may be distributed under.

-----  
GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's

software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this

License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty;

and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program



with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License.

However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed

through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,

REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice

like this  
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of  
Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

## 1.126 ngrep 1.47-r4

### 1.126.1 Available under license :

GNU GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
675 Mass Ave, Cambridge, MA 02139, USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This

General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole

or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an

announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you



received the program in object code  
or executable form with such  
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy  
the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6.

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program

specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Copyright (c) 2017 Jordan Ritter. All rights reserved.

Permission is granted to anyone to use this software for any purpose on any computer system, and to alter it and redistribute it, subject to the following restrictions:

1. The origin of this software must not be misrepresented, either by explicit claim or by omission.
2. Altered versions must be plainly marked as such, and must not be misrepresented as being the original software. Any altered version must clearly and properly represent the origin of this software in any accompanying documentation.
3. All advertising materials which relate specifically to derivate works of this software must display the following acknowledgement:  
This product includes software developed by Jordan Ritter.
4. The name of the Author may not be used to endorse or promote products derived from this software without specific prior written permission.
5. This notice, and any references to this notice, in any original or derived source distribution of or documentation for this software, may not be removed or altered.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions were copied from tcpkill (part of dsniff), which has the following copyright:

Copyright (c) 1999, 2000 Dug Song <dugsong@monkey.org>  
All rights reserved, all wrongs reversed.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.127 xz 5.8.1-r0

### 1.127.1 Available under license :

Copyright (c) 2014-2020 Ulrich Kunitz  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\* My name, Ulrich Kunitz, may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR

ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Copyright 2014-2019 Ulrich Kunitz. All rights reserved.

// Use of this source code is governed by a BSD-style

// license that can be found in the LICENSE file.

package main

```
import (  
    "bufio"  
    "bytes"  
    "flag"  
    "fmt"  
    "io"  
    "log"  
    "os"  
    "path/filepath"  
    "strings"  
)
```

```
const crUsageString = `xb copyright [options] <path>....
```

The xb copyright command adds a copyright remark to all go files below path.

```
-h prints this message and exits  
`
```

```
func crUsage(w io.Writer) {  
    fmt.Fprint(w, crUsageString)  
}
```

```
const copyrightText = `  
Copyright 2014-2020 Ulrich Kunitz. All rights reserved.  
Use of this source code is governed by a BSD-style  
license that can be found in the LICENSE file.  
`
```

```

func goComment(text string) string {
    buf := new(bytes.Buffer)
    scanner := bufio.NewScanner(strings.NewReader(text))
    var err error
    for scanner.Scan() {
        s := strings.TrimSpace(scanner.Text())
        if len(s) == 0 {
            continue
        }
        if _, err = fmt.Fprintln(buf, "///", s); err != nil {
            panic(err)
        }
    }
    if err
    = scanner.Err(); err != nil {
        panic(err)
    }
    if _, err = fmt.Fprintln(buf); err != nil {
        panic(err)
    }
    return buf.String()
}

```

```

var goCopyright = goComment(copyrightText)

```

```

func addCopyright(path string) (err error) {
    log.Printf("adding copyright to %s", path)
    src, err := os.Open(path)
    if err != nil {
        return err
    }
    defer func() {
        cerr := src.Close()
        if cerr != nil && err == nil {
            err = cerr
        }
    }()
    newPath := path + ".new"
    dst, err := os.Create(newPath)
    if err != nil {
        return err
    }
    defer func() {
        cerr := dst.Close()
        if cerr != nil && err == nil {
            err = cerr
        }
    }
}

```



```

}()
out := bufio.NewWriter(dst)
fmt.Fprint(out, goCopyright)
scanner := bufio.NewScanner(src)
line := 0
del := false
for scanner.Scan() {
    line++
    txt := scanner.Text()
    if line == 1 && strings.Contains(txt, "Copyright") {
        del = true
        continue
    }
    if del {
        s := strings.TrimSpace(txt)
        if len(s) == 0 {
            del = false
        }
        continue
    }
    fmt.Fprintln(out, txt)
}
if err = scanner.Err();
err != nil {
    return err
}
if err = out.Flush(); err != nil {
    return
}
err = os.Rename(newPath, path)
return
}

func walkCopyrights(path string, info os.FileInfo, err error) error {
    if err != nil {
        return err
    }
    if info.IsDir() {
        return nil
    }
    if !strings.HasSuffix(info.Name(), ".go") {
        return nil
    }
    return addCopyright(path)
}

func copyright() {
    cmdName := os.Args[0]

```

```

log.SetPrefix(fmt.Sprintf("%s: ", cmdName))
log.SetFlags(0)

flag.CommandLine = flag.NewFlagSet(cmdName, flag.ExitOnError)
flag.Usage = func() { crUsage(os.Stderr); os.Exit(1) }

help := flag.Bool("h", false, "")

flag.Parse()

if *help {
    crUsage(os.Stdout)
    os.Exit(0)
}

for _, path := range flag.Args() {
    fi, err := os.Stat(path)
    if err != nil {
        log.Print(err)
        continue
    }
    if !fi.IsDir() {
        log.Printf("%s is not a directory", path)
        continue
    }
    if err = filepath.Walk(path, walkCopyrights); err != nil {
        log.Fatalf("%s error %s", path, err)
    }
}
}
package main

```

const goLicense = `Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

const xzLicense = `Copyright (c) 2014-2020 Ulrich Kunitz  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* My name, Ulrich Kunitz, may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.128 py3-parsing 3.2.3-r0

## 1.128.1 Available under license :

MIT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.129 lua5.4 5.4.7-r0

### 1.129.1 Available under license :

No license file was found, but licenses were detected in source scan.

@Ci{\$Id: manual.of \$}

@C{[(-----)}

@manual{

@sect1{@title{Introduction}}

Lua is a powerful, efficient, lightweight, embeddable scripting language. It supports procedural programming, object-oriented programming, functional programming, data-driven programming, and data description.

Lua combines simple procedural syntax with powerful data description constructs based on associative arrays and extensible semantics.

Lua is dynamically typed, runs by interpreting bytecode with a register-based virtual machine, and has automatic memory management with a generational garbage collection, making it ideal for configuration, scripting, and rapid prototyping.

Lua is implemented as a library, written in `@emphx{clean C}`,  
the common subset of `@N{standard C}` and C++.

The Lua distribution includes a host program called `@id{lua}`,  
which uses the Lua library to offer  
a complete,  
standalone Lua interpreter,  
for interactive or batch use.

Lua is intended to be used both as a powerful, lightweight,  
embeddable scripting language for any program that needs one,  
and as a powerful but lightweight and efficient stand-alone language.

As an extension language, Lua has no notion of a `@Q{main}` program:  
it works `@emph{embedded}` in a host client,  
called the `@emph{embedding program}` or simply the `@emphx{host}`.  
(Frequently, this host is the stand-alone `@id{lua}` program.)

The host program can invoke functions to execute a piece of Lua code,  
can write and read Lua variables,  
and can register `@N{C functions}` to be called by Lua code.

Through the use of `@N{C functions}`, Lua can be augmented to cope with  
a wide range of different domains,  
thus creating customized programming languages sharing a syntactical framework.

Lua is free software,  
and is provided as usual with no guarantees,  
as stated in its license.

The implementation described in this manual is available  
at  
Lua's official web site, `@id{www.lua.org}`.

Like any other reference manual,  
this document is dry in places.

For a discussion of the decisions behind the design of Lua,  
see the technical papers available at Lua's web site.

For a detailed introduction to programming in Lua,  
see Roberto's book, `@emphx{Programming in Lua}`.

}

`@C{-----}`  
`@sect1{basic| @title{Basic Concepts}`

`@simplesect{`

This section describes the basic concepts of the language.

}

Lua is a dynamically typed language.

This means that

variables do not have types; only values do.

There are no type definitions in the language.

All values carry their own type.

All values in Lua are first-class values.

This means that all values can be stored in variables,  
passed as arguments to other functions, and returned as results.

There are eight @x{basic types} in Lua:

@def{nil}, @def{boolean}, @def{number},

@def{string},

@def{function}, @def{userdata},

@def{thread}, and @def{table}.

The type @emph{nil} has one single value, @nil,  
whose main property is to be different from any other value;  
it often represents the absence of a useful value.

The type @emph{boolean} has two values, @false and @true.

Both @nil and @false make a condition false;  
they are collectively called @def{false values}.

Any other value makes a condition true.

Despite its name,

@false is frequently used as an alternative to @nil,  
with the key difference that @false behaves  
like a regular value in a table,  
while a @nil in a table represents an absent key.

The type @emph{number} represents both  
integer numbers and real (floating-point) numbers,  
using two @x{subtypes}: @def{integer} and @def{float}.

Standard Lua uses 64-bit integers and double-precision (64-bit) floats,  
but you can also compile Lua so that it  
uses 32-bit integers and/or single-precision (32-bit) floats.

The option with 32 bits for both integers and floats  
is particularly

attractive

for small machines and embedded systems.

(See macro @id{LUA\_32BITS} in file @id{luaconf.h}.)

Unless stated otherwise,

any overflow when manipulating integer values @def{wrap around},  
according to the usual rules of two-complement arithmetic.

(In other words,

the actual result is the unique representable integer

that is equal modulo  $2^M$  to the mathematical result,  
where  $M$  is the number of bits of the integer type.)

Lua has explicit rules about when each subtype is used,  
but it also converts between them automatically as needed (see coercion).  
Therefore,  
the programmer may choose to mostly ignore the difference  
between integers and floats  
or to assume complete control over the representation of each number.

The type `string` represents immutable sequences of bytes.  
(Eight-bit clean)

Lua is 8-bit clean:

strings can contain any 8-bit value,  
including embedded zeros (`\0`).

Lua is also encoding-agnostic;  
it makes no assumptions about the contents  
of a string.

The length of any string in Lua must fit in a Lua integer.

Lua can call (and manipulate) functions written in Lua and  
functions written in C (see function call).

Both are represented by the type `function`.

The type `userdata` is provided to allow arbitrary C data to  
be stored in Lua variables.

A userdata value represents a block of raw memory.

There are two kinds of userdata:

`full userdata`,

which is an object with a block of memory managed by Lua,

and `light userdata`,

which is simply a C pointer value.

Userdata has no predefined operations in Lua,  
except assignment and identity test.

By using `metatables`,  
the programmer can define operations for full userdata values  
(see metatable).

Userdata values cannot be created or modified in Lua,  
only through the C API.

This guarantees the integrity of data owned by  
the host program and C libraries.

The type `thread` represents independent threads of execution  
and it is used to  
implement coroutines (see coroutine).

Lua threads are not related to operating-system threads.

Lua supports coroutines on all systems,  
even those that do not support threads natively.

The type `@emph{table}` implements `@x{associative arrays}`, that is, `@x{arrays}` that can have as indices not only numbers, but any Lua value except `@nil` and `@x{NaN}`. (`@emphx{Not a Number}` is a special floating-point value used by the `@x{IEEE 754}` standard to represent undefined numerical results, such as `@T{0/0}`.) Tables can be `@emph{heterogeneous}`; that is, they can contain values of all types (except `@nil`). Any key associated to the value `@nil` is not considered part of the table. Conversely, any key that is not part of a table has an associated value `@nil`.

Tables are the sole data-structuring mechanism in Lua; they can be used to represent ordinary arrays, lists, symbol tables, sets, records, graphs, trees, etc. To represent `@x{records}`, Lua uses the field name as an index. The language supports this representation by providing `@id{a.name}` as syntactic sugar for `@T{a["name"]}`. There are several convenient ways to create tables in Lua `@see{tableconstructor}`.

Like indices, the values of table fields can be of any type. In particular, because functions are first-class values, table fields can contain functions. Thus tables can also carry `@emph{methods}` `@see{func-def}`.

The indexing of tables follows the definition of raw equality in the language. The expressions `@T{a[i]}` and `@T{a[j]}` denote the same table element if and only if `@id{i}` and `@id{j}` are raw equal (that is, equal without metamethods). In particular, floats with integral values are equal to their respective integers (e.g., `@T{1.0 == 1}`). To avoid ambiguities, any float used as a key that is equal to an integer is converted to that integer. For instance, if you write `@T{a[2.0] = true}`, the actual key inserted into the table will be the integer `@T{2}`.

Tables, functions, threads, and (full) userdata values are `@emph{objects}`: variables



do not actually @emph{contain} these values,  
only @emph{references} to them.  
Assignment, parameter passing, and function returns  
always manipulate references to such values;  
these operations do not imply any kind of copy.

The library function @Lid{type} returns a string describing the type  
of a given value @seeF{type}.

}

@sect2{globalenv| @title{Environments and the Global Environment}}

As we will discuss further in @refsec{variables} and @refsec{assignment},  
any reference to a free name  
(that is, a name not bound to any declaration) @id{var}  
is syntactically translated to @T{\_ENV.var}.  
Moreover, every chunk is compiled in the scope of  
an external local variable named @id{\_ENV} @see{chunks},  
so @id{\_ENV} itself is never a free name in a chunk.

Despite the existence of this external @id{\_ENV} variable and  
the translation of free names,  
@id{\_ENV} is a completely regular name.  
In particular,  
you can define new variables and parameters with that name.  
Each reference to a free  
name uses the @id{\_ENV} that is  
visible at that point in the program,  
following the usual visibility rules of Lua @see{visibility}.

Any table used as the value of @id{\_ENV} is called an @def{environment}.

Lua keeps a distinguished environment called the @def{global environment}.  
This value is kept at a special index in the C registry @see{registry}.  
In Lua, the global variable @Lid{\_G} is initialized with this same value.  
(@Lid{\_G} is never used internally,  
so changing its value will affect only your own code.)

When Lua loads a chunk,  
the default value for its @id{\_ENV} variable  
is the global environment @seeF{load}.  
Therefore, by default,  
free names in Lua code refer to entries in the global environment  
and, therefore, they are also called @def{global variables}.  
Moreover, all standard libraries are loaded in the global environment  
and some functions there operate on that environment.  
You can use @Lid{load} (or @Lid{loadfile})

to load a chunk with a different environment.  
(In C,  
you have to load the chunk and then change the value  
of its first upvalue; see @See{lua\_setupvalue}.)

```
}
```

@sect2{error| @title{Error Handling}}

Several operations in Lua can @emph{raise} an error.  
An error interrupts the normal flow of the program,  
which can continue by @emph{catching} the error.

Lua code can explicitly raise an error by calling the  
@Lid{error} function.  
(This function never returns.)

To catch errors in Lua,  
you can do a @def{protected call},  
using @Lid{pcall} (or @Lid{xpcall}).  
The function @Lid{pcall} calls a given function in @def{protected mode}.  
Any error while running the function stops its execution,  
and control returns immediately to @id{pcall},  
which returns a status code.

Because Lua is an embedded extension language,  
Lua code starts running by a call  
from @N{C code} in the host program.  
(When you use Lua standalone,  
the @id{lua} application is the host program.)  
Usually, this call is protected;  
so, when an otherwise unprotected error occurs during  
the compilation  
or execution of a Lua chunk,  
control returns to the host,  
which can take appropriate measures,  
such as printing an error message.

Whenever there is an error,  
an @def{error object}  
is propagated with information about the error.  
Lua itself only generates errors whose error object is a string,  
but programs can generate errors with  
any value as the error object.  
It is up to the Lua program or its host to handle such error objects.  
For historical reasons,  
an error object is often called an @def{error message},  
even though it does not have to be a string.

When you use `@Lid{xpcall}` (or `@Lid{lua_pcall}`, in C) you can give a `@def{message handler}` to be called in case of errors. This function is called with the original error object and returns a new error object. It is called before the error unwinds the stack, so that it can gather more information about the error, for instance by inspecting the stack and creating a stack traceback. This message handler is still protected by the protected call; so, an error inside the message handler will call the message handler again. If this loop goes on for too long, Lua breaks it and returns an appropriate message. The message handler is called only for regular runtime errors. It is not called for memory-allocation errors nor for errors while running finalizers or other message handlers.

Lua also offers a system of `@emph{warnings}` `@seeF{warn}`. Unlike errors, warnings do not interfere in any way with program execution. They typically only generate a message to the user, although this behavior can be adapted from C `@seeC{lua_setwarnf}`.

}

`@sect2{metatable| @title{Metatables and Metamethods}}`

Every value in Lua can have a `@emph{metatable}`. This `@def{metatable}` is an ordinary Lua table that defines the behavior of the original value under certain events. You can change several aspects of the behavior of a value by setting specific fields in its metatable. For instance, when a non-numeric value is the operand of an addition, Lua checks for a function in the field `@idx{__add}` of the value's metatable. If it finds one, Lua calls this function to perform the addition.

The key for each event in a metatable is a string with the event name prefixed by two underscores; the corresponding value is called a `@def{metavalue}`. For most events, the metavalue must be a function, which is then called a `@def{metamethod}`. In the previous example, the key is the string `@St{__add}`

and the metamethod is the function that performs the addition.  
Unless stated otherwise,  
a metamethod can in fact be any `@x{callable value}`,  
which is either a function or a value with a `@idx{__call}` metamethod.

You can query the metatable of any value  
using the `@Lid{getmetatable}` function.  
Lua queries metamethods in metatables using a raw access `@seeF{rawget}`.

You can replace the metatable of tables  
using the `@Lid{setmetatable}` function.  
You cannot change the metatable of other types from Lua code,  
except by using the `@link{debuglib|debug library}`.

Tables  
and full userdata have individual metatables,  
although multiple tables and userdata can share their metatables.  
Values of all other types share one single metatable per type;  
that is, there is one single metatable for all numbers,  
one for all strings, etc.  
By default, a value has no metatable,  
but the string library sets a metatable for the string type `@see{strlib}`.

A detailed list of operations controlled by metatables is given next.  
Each event is identified by its corresponding key.  
By convention, all metatable keys used by Lua are composed by  
two underscores followed by lowercase Latin letters.

`@description{`

`@item{@idx{__add}|`  
the addition (`@T{+}`) operation.  
If any operand for an addition is not a number,  
Lua will try to call a metamethod.  
It starts by checking the first operand (even if it is a number);  
if that operand does not define a metamethod for `@idx{__add}`,  
then Lua will check the second operand.  
If Lua can find a metamethod,  
it calls the metamethod with the two  
operands as arguments,  
and the result of the call  
(adjusted to one value)  
is the result of the operation.  
Otherwise, if no metamethod is found,  
Lua raises an error.  
`}`

`@item{@idx{__sub}|`

the subtraction ( $@T\{-}$ ) operation.  
Behavior similar to the addition operation.  
}

@item{@idx{\_\_mul}|  
the multiplication ( $@T\{*\}$ ) operation.  
Behavior similar to the addition operation.  
}

@item{@idx{\_\_div}|  
the division ( $@T\{/}$ ) operation.  
Behavior similar to the addition operation.  
}

@item{@idx{\_\_mod}|  
the modulo ( $@T\{\%\}$ ) operation.  
Behavior similar to the addition operation.  
}

@item{@idx{\_\_pow}|  
the exponentiation ( $@T\{^\wedge\}$ ) operation.  
Behavior similar to the addition operation.  
}

@item{@idx{\_\_unm}|  
the negation (unary  $@T\{-}$ ) operation.  
Behavior similar to the addition operation.  
}

@item{@idx{\_\_idiv}|  
the floor division ( $@T\{//}$ ) operation.  
Behavior similar to the addition operation.  
}

@item{@idx{\_\_band}|  
the bitwise AND ( $@T\{\&\}$ ) operation.  
Behavior similar to the addition operation,  
except that Lua will  
try a metamethod  
if any operand is neither an integer  
nor a float coercible to an integer @see{coercion}.  
}

@item{@idx{\_\_bor}|  
the bitwise OR ( $@T\{\|\}$ ) operation.  
Behavior similar to the bitwise AND operation.  
}

@item{@idx{\_\_bxor}|  
the bitwise exclusive OR (binary @T{~}) operation.  
Behavior similar to the bitwise AND operation.  
}

@item{@idx{\_\_bnot}|  
the bitwise NOT (unary @T{~}) operation.  
Behavior similar to the bitwise AND operation.  
}

@item{@idx{\_\_shl}|  
the bitwise left shift (@T{<<}) operation.  
Behavior similar to the bitwise AND operation.  
}

@item{@idx{\_\_shr}|  
the bitwise right shift (@T{>>}) operation.  
Behavior similar to the bitwise AND operation.  
}

@item{@idx{\_\_concat}|  
the concatenation (@T{..}) operation.  
Behavior similar to the addition operation,  
except that Lua will try a metamethod  
if any operand is neither a string nor a number  
(which is always coercible to a string).  
}

@item{@idx{\_\_len}|  
the length (@T{#}) operation.  
If the object is not a string,  
Lua will try its  
metamethod.  
If there is a metamethod,  
Lua calls it with the object as argument,  
and the result of the call  
(always adjusted to one value)  
is the result of the operation.  
If there is no metamethod but the object is a table,  
then Lua uses the table length operation @see{len-op}.  
Otherwise, Lua raises an error.  
}

@item{@idx{\_\_eq}|  
the equal (@T{==}) operation.  
Behavior similar to the addition operation,  
except that Lua will try a metamethod only when the values  
being compared are either both tables or both full userdata

and they are not primitively equal.  
The result of the call is always converted to a boolean.  
}

@item{@idx{\_\_lt}|  
the less than (@T{<}) operation.  
Behavior similar to the addition operation,  
except that Lua will try a metamethod only when the values  
being compared are neither both numbers nor both strings.  
Moreover, the result of the call is always converted to a boolean.  
}

@item{@idx{\_\_le}|  
the less equal (@T{<=}) operation.  
Behavior similar to the less than operation.  
}

@item{@idx{\_\_index}|  
The  
indexing access operation @T{table[key]}.  
This event happens when @id{table} is not a table or  
when @id{key} is not present in @id{table}.  
The metavalue is looked up in the metatable of @id{table}.

The metavalue for this event can be either a function, a table,  
or any value with an @idx{\_\_index} metavalue.  
If it is a function,  
it is called with @id{table} and @id{key} as arguments,  
and the result of the call  
(adjusted to one value)  
is the result of the operation.  
Otherwise,  
the final result is the result of indexing this metavalue with @id{key}.  
This indexing is regular, not raw,  
and therefore can trigger another @idx{\_\_index} metavalue.  
}

@item{@idx{\_\_newindex}|  
The indexing assignment @T{table[key] = value}.  
Like the index event,  
this event happens when @id{table} is not a table or  
when @id{key} is not present in @id{table}.  
The metavalue is looked up in the metatable of @id{table}.

Like with indexing,  
the metavalue for this event can be either a function,  
a table,  
or any value with an @idx{\_\_newindex} metavalue.

If it is a function,  
it is called with `@id{table}`, `@id{key}`, and `@id{value}` as arguments.  
Otherwise,  
Lua repeats the indexing assignment over this metavalue  
with the same key and value.  
This assignment is regular, not raw,  
and therefore can trigger another `@idx{__newindex}` metavalue.

Whenever a `@idx{__newindex}` metavalue is invoked,  
Lua does not perform the primitive assignment.  
If needed,  
the metamethod itself can call `@Lid{rawset}`  
to do the assignment.  
}

`@item{@idx{__call}|`  
The call operation `@T{func(args)}`.  
This event happens when Lua tries to call a non-function value  
(that is, `@id{func}` is not a function).  
The metamethod is looked up in `@id{func}`.  
If present,  
the metamethod is called with `@id{func}` as its first argument,  
followed by the arguments of the original call (`@id{args}`).  
All results of the call  
are the results of the operation.  
This is the only metamethod that allows multiple results.  
}  
}

In addition to  
the previous list,  
the interpreter also respects the following keys in metatables:  
`@idx{__gc}` `@see{finalizers}`,  
`@idx{__close}` `@see{to-be-closed}`,  
`@idx{__mode}` `@see{weak-table}`,  
and `@idx{__name}`.  
(The entry `@idx{__name}`,  
when it contains a string,  
may be used by `@Lid{tostring}` and in error messages.)

For the unary operators (negation, length, and bitwise NOT),  
the metamethod is computed and called with a dummy second operand,  
equal to the first one.  
This extra operand is only to simplify Lua's internals  
(by making these operators behave like a binary operation)  
and may be removed in future versions.  
For most uses this extra operand is irrelevant.



Because metatables are regular tables,  
they can contain arbitrary fields,  
not only the event names defined above.  
Some functions in the standard library  
(e.g., `@Lid{tostring}`)  
use other fields in metatables for their own purposes.

It is a good practice to add all needed metamethods to a table  
before setting it as a metatable of some object.

In

particular, the `@idx{__gc}` metamethod works only when this order  
is followed `@see{finalizers}`.

It is also a good practice to set the metatable of an object  
right after its creation.

}

`@sect2{GC| @title{Garbage Collection}`

`@simplesect{`

Lua performs automatic memory management.

This means that

you do not have to worry about allocating memory for new objects  
or freeing it when the objects are no longer needed.

Lua manages memory automatically by running  
a `@def{garbage collector}` to collect all `@emph{dead}` objects.

All memory used by Lua is subject to automatic management:  
strings, tables, userdata, functions, threads, internal structures, etc.

An object is considered `@def{dead}`  
as soon as the collector can be sure the object  
will not be accessed again in the normal execution of the program.  
(`@Q{Normal execution}` here excludes finalizers,  
which can resurrect dead objects `@see{finalizers}`,  
and excludes also operations using the debug library.)

Note that the time when the collector can  
be sure that an object

is dead may not coincide with the programmer's expectations.

The only guarantees are that Lua will not collect an object  
that may still be accessed in the normal execution of the program,  
and it will eventually collect an object  
that is inaccessible from Lua.

(Here,

`@emph{inaccessible from Lua}` means that neither a variable nor  
another live object refer to the object.)

Because Lua has no knowledge about `@N{C code}`,

it never collects objects accessible through the registry `@see{registry}`, which includes the global environment `@see{globalenv}`.

The garbage collector (GC) in Lua can work in two modes: incremental and generational.

The default GC mode with the default parameters are adequate for most uses.

However, programs that waste a large proportion of their time allocating and freeing memory can benefit from other settings.

Keep in mind that the GC behavior is non-portable both across platforms and across different Lua releases; therefore, optimal settings are also non-portable.

You can change the GC mode and parameters by calling

`@Lid{lua_gc}` `@N{in C}`

or `@Lid{collectgarbage}` in Lua.

You can also use these functions to control the collector directly (e.g., to stop and restart it).

}

`@sect3{incmode| @title{Incremental Garbage Collection}`

In incremental mode, each GC cycle performs a mark-and-sweep collection in small steps interleaved with the program's execution.

In this mode,

the collector uses three numbers to control its garbage-collection cycles:

the `@def{garbage-collector pause}`,

the `@def{garbage-collector step multiplier}`,

and the `@def{garbage-collector step size}`.

The garbage-collector pause

controls how long the collector waits before starting a new cycle.

The collector starts a new cycle when the use of memory hits `@M{n%}` of the use after the previous collection.

Larger values make the collector less aggressive.

Values equal to or less than 100 mean the collector will not wait to start a new cycle.

A value of 200 means that

the collector waits for the total memory in use to double before starting a new cycle.

The default value is 200; the maximum value is 1000.

The garbage-collector step multiplier

controls the speed of the collector relative to memory allocation, that is, how many elements it marks or sweeps for each kilobyte of memory allocated. Larger values make the collector more aggressive but also increase the size of each incremental step. You should not use values less than 100, because they make the collector too slow and can result in the collector never finishing a cycle. The default value is 100; the maximum value is 1000.

The garbage-collector step size controls the size of each incremental step, specifically how many bytes the interpreter allocates before performing a step.

This parameter is logarithmic:

A value of  $M_n$  means the interpreter will allocate  $M_{2^n}$  bytes between steps and perform equivalent work during the step.

A large value (e.g., 60) makes the collector a stop-the-world (non-incremental) collector.

The default value is 13, which means steps of approximately 8 Kbytes.

}

@sect3{genmode| @title{Generational Garbage Collection}

In generational mode, the collector does frequent *minor* collections, which traverses only objects recently created. If after a minor collection the use of memory is still above a limit, the collector does a stop-the-world *major* collection, which traverses all objects.

The generational mode uses two parameters: the *minor multiplier* and the *major multiplier*.

The minor multiplier controls the frequency of minor collections. For a minor multiplier  $M_x$ , a new minor collection will be done when memory grows  $M_x\%$  larger than the memory in use after the previous major collection.

For instance, for a multiplier of 20, the collector will do a minor collection when the use of memory gets 20% larger than the use after the previous major collection. The default value is 20; the maximum value is 200.

The major multiplier controls the frequency of major collections. For a major multiplier `@M{x}`, a new major collection will be done when memory grows `@M{x%}` larger than the memory in use after the previous major collection. For instance, for a multiplier of 100, the collector will do a major collection when the use of memory gets larger than twice the use after the previous collection. The default value is 100; the maximum value is 1000.

}

@sect3{finalizers| @title{Garbage-Collection Metamethods}

You can set garbage-collector metamethods for tables and, using the `@N{C API}`, for full userdata `@see{metatable}`. These metamethods, called `@def{finalizers}`, are called when the garbage collector detects that the corresponding table or userdata is dead. Finalizers allow you to coordinate Lua's garbage collection with external resource management such as closing files, network or database connections, or freeing your own memory.

For an object (table or userdata) to be finalized when collected, you must `@emph{mark}` it for finalization. `@index{mark (for finalization)}` You mark an object for finalization when you set its metatable and the metatable has a `@idx{__gc}` metamethod. Note that if you set a metatable without a `@idx{__gc}` field and later create that field in the metatable, the object will not be marked for finalization.

When a marked object becomes dead, it is not collected immediately by the garbage collector. Instead, Lua puts it in a list. After the collection, Lua goes through that list. For each object in the list, it checks the object's `@idx{__gc}` metamethod: If it is present, Lua calls it with the object as its single argument.

At the end of each garbage-collection cycle, the finalizers are called in

the reverse order that the objects were marked for finalization,  
among those collected in that cycle;  
that is, the first finalizer to be called is the one associated  
with the object marked last in the program.  
The execution of each finalizer may occur at any  
point during  
the execution of the regular code.

Because the object being collected must still be used by the finalizer,  
that object (and other objects accessible only through it)  
must be *@emph{resurrected}* by Lua.*@index{resurrection}*  
Usually, this resurrection is transient,  
and the object memory is freed in the next garbage-collection cycle.  
However, if the finalizer stores the object in some global place  
(e.g., a global variable),  
then the resurrection is permanent.  
Moreover, if the finalizer marks a finalizing object for finalization again,  
its finalizer will be called again in the next cycle where the  
object is dead.  
In any case,  
the object memory is freed only in a GC cycle where  
the object is dead and not marked for finalization.

When you close a state *@seeF{lua\_close}*,  
Lua calls the finalizers of all objects marked for finalization,  
following the reverse order that they were marked.  
If any finalizer marks objects for collection during that phase,  
these marks have no effect.

Finalizers  
cannot yield nor run the garbage collector.  
Because they can run in unpredictable times,  
it is good practice to restrict each finalizer  
to the minimum necessary to properly release  
its associated resource.

Any error while running a finalizer generates a warning;  
the error is not propagated.

}

*@sect3{weak-table| @title{Weak Tables}*

A *@def{weak table}* is a table whose elements are  
*@def{weak references}*.  
A weak reference is ignored by the garbage collector.  
In other words,  
if the only references to an object are weak references,

then the garbage collector will collect that object.

A weak table can have weak keys, weak values, or both.

A table with weak values allows the collection of its values, but prevents the collection of its keys.

A table with both weak keys and weak values allows the collection of both keys and values.

In any case, if either the key or the value is collected, the whole pair is removed from the table.

The weakness of a table is controlled by the

`@idx{__mode}`

field of its metatable.

This metavalue, if present, must be one of the following strings:

`@St{k}`, for a table with weak keys;

`@St{v}`, for a table with weak values;

or `@St{kv}`, for a table with both weak keys and values.

A table with weak keys and strong values is also called an `@def{ephemeron table}`.

In an ephemeron table,

a value is considered reachable only if its key is reachable.

In particular,

if the only reference to a key comes through its value, the pair is removed.

Any change in the weakness of a table may take effect only at the next collect cycle.

In particular, if you change the weakness to a stronger mode,

Lua may still collect some items from that table

before the change takes effect.

Only objects that have an explicit construction are removed from weak tables.

Values, such as numbers and `@x{light @N{C functions}}`,

are not subject to garbage collection,

and therefore are not removed from weak tables

(unless their associated values are collected).

Although strings

are subject to garbage collection,

they do not have an explicit construction and

their equality is by value;

they behave more like values than like objects.

Therefore, they are not removed from weak tables.

Resurrected objects

(that is, objects being finalized

and objects accessible only through objects being finalized)

have a special behavior in weak tables.

They are removed from weak values before running their finalizers, but are removed from weak keys only in the next collection after running their finalizers, when such objects are actually freed. This behavior allows the finalizer to access properties associated with the object through weak tables.

If a weak table is among the resurrected objects in a collection cycle, it may not be properly cleared until the next cycle.

}

}

@sect2{coroutine| @title{Coroutines}

Lua supports coroutines, also called @emphx{collaborative multithreading}. A coroutine in Lua represents an independent thread of execution. Unlike threads in multithread systems, however, a coroutine only suspends its execution by explicitly calling a yield function.

You create a coroutine by calling @Lid{coroutine.create}. Its sole argument is a function that is the main function of the coroutine. The @id{create} function only creates a new coroutine and returns a handle to it (an object of type @emph{thread}); it does not start the coroutine.

You execute a coroutine by calling @Lid{coroutine.resume}. When you first call @Lid{coroutine.resume}, passing as its first argument a thread returned by @Lid{coroutine.create}, the coroutine starts its execution by calling its main function. Extra arguments passed to @Lid{coroutine.resume} are passed as arguments to that function. After the coroutine starts running, it runs until it terminates or @emph{yields}.

A coroutine can terminate its execution in two ways: normally, when its main function returns (explicitly or implicitly, after the last instruction); and abnormally, if there is an unprotected error.

In case of normal termination, @Lid{coroutine.resume} returns @true,

plus any values returned by the coroutine main function.  
In case of errors, `@Lid{coroutine.resume}` returns `@false`  
plus the error object.  
In this case, the coroutine does not unwind its stack,  
so that it is possible to inspect it after the error  
with the debug API.

A coroutine yields by calling `@Lid{coroutine.yield}`.  
When a coroutine yields,  
the corresponding `@Lid{coroutine.resume}` returns immediately,  
even if the yield happens inside nested function calls  
(that is, not in the main function,  
but in a function directly or indirectly called by the main function).  
In the case of a yield, `@Lid{coroutine.resume}` also returns `@true`,  
plus any values passed to `@Lid{coroutine.yield}`.  
The next time you resume the same coroutine,  
it continues its execution from the point where it yielded,  
with the call to `@Lid{coroutine.yield}` returning any extra  
arguments passed to `@Lid{coroutine.resume}`.

Like `@Lid{coroutine.create}`,  
the  
`@Lid{coroutine.wrap}` function also creates a coroutine,  
but instead of returning the coroutine itself,  
it returns a function that, when called, resumes the coroutine.  
Any arguments passed to this function  
go as extra arguments to `@Lid{coroutine.resume}`.  
`@Lid{coroutine.wrap}` returns all the values returned by `@Lid{coroutine.resume}`,  
except the first one (the boolean error code).  
Unlike `@Lid{coroutine.resume}`,  
the function created by `@Lid{coroutine.wrap}`  
propagates any error to the caller.  
In this case,  
the function also closes the coroutine `@seeF{coroutine.close}`.

As an example of how coroutines work,  
consider the following code:

```
@verbatim{
function foo (a)
  print("foo", a)
  return coroutine.yield(2*a)
end

co = coroutine.create(function (a,b)
  print("co-body", a, b)
  local r = foo(a+1)
  print("co-body", r)
  local r, s = coroutine.yield(a+b, a-b)
```



```

    print("co-body", r, s)
    return b, "end"
end)

print("main", coroutine.resume(co, 1, 10))
print("main", coroutine.resume(co,
    "r"))
print("main", coroutine.resume(co, "x", "y"))
print("main", coroutine.resume(co, "x", "y"))
}

```

When you run it, it produces the following output:

```

@verbatim{
co-body 1    10
foo    2
main  true  4
co-body r
main  true  11  -9
co-body x    y
main  true  10  end
main  false  cannot resume dead coroutine
}

```

You can also create and manipulate coroutines through the C API:

see functions `@Lid{lua_newthread}`, `@Lid{lua_resume}`,  
and `@Lid{lua_yield}`.

```

}

```

```

}

```

```

@C{-----}
@sect1{language| @title{The Language}

```

```

@simplesect{

```

This section describes the lexis, the syntax, and the semantics of Lua.

In other words,

this section describes

which tokens are valid,

how they can be combined,

and what their combinations mean.

Language constructs will be explained using the usual extended BNF notation,  
in which

`@N{@bnfrep{@rep{a}}` means 0} or more `@rep{a}`'s, and

`@N{@bnfopt{@rep{a}}` means} an

optional @rep{a}.

Non-terminals are shown like @bnfNter{non-terminal},

keywords are shown like @rw{kword},

and other terminal symbols are shown like @bnfter{=}.

The complete syntax of Lua can be found in @refsec{BNF}

at the end of this manual.

}

@sect2{lexical| @title{Lexical Conventions}}

Lua is a @x{free-form} language.

It ignores spaces and comments between lexical elements (@x{tokens}),

except as delimiters between two tokens.

In source code,

Lua recognizes as spaces the standard ASCII whitespace

characters space, form feed, newline,

carriage return, horizontal tab, and vertical tab.

@def{Names}

(also called @def{identifiers})

in Lua can be any string of Latin letters,

Arabic-Indic digits, and underscores,

not beginning with a digit and

not being a reserved word.

Identifiers are used to name variables, table fields, and labels.

The following @def{keywords} are reserved

and cannot be used as names:

@index{reserved words}

@verbatim{

and    break    do    else    elseif

      end

false   for    function goto   if    in

local   nil    not    or    repeat   return

then    true    until   while

}

Lua is a case-sensitive language:

@id{and} is a reserved word, but @id{And} and @id{AND}

are two different, valid names.

As a convention,

programs should avoid creating

names that start with an underscore followed by

one or more uppercase letters (such as @Lid{\_VERSION}).

The following strings denote other @x{tokens}:

@verbatim{

```

+ - * / % ^ #
& ~ | << >> //
== ~= <= >= < > =
( ) { } [ ] ::
; : , . .. ...
}

```

A `@def{short literal string}`

can be delimited by matching single or double quotes,  
and can contain the following C-like escape sequences:

```

@Char{a} (bell),
@Char{b} (backspace),
@Char{f} (form feed),
@Char{n} (newline),
@Char{r} (carriage return),
@Char{t} (horizontal tab),
@Char{v} (vertical tab),
@Char{\\} (backslash),
@Char{"}
    (quotation mark [double quote]),
and @Char{'} (apostrophe [single quote]).

```

A backslash followed by a line break  
results in a newline in the string.

The escape sequence `@Char{z}` skips the following span  
of whitespace characters,  
including line breaks;

it is particularly useful to break and indent a long literal string  
into multiple lines without adding the newlines and spaces  
into the string contents.

A short literal string cannot contain unescaped line breaks  
nor escapes not forming a valid escape sequence.

We can specify any byte in a short literal string,  
including `@x{embedded zeros}`,  
by its numeric value.

This can be done

with the escape sequence `@T{x@rep{XX}}`,  
where `@rep{XX}` is a sequence of exactly two hexadecimal digits,  
or with the escape sequence `@T{@rep{ddd}}`,  
where `@rep{ddd}` is a sequence of up to three decimal digits.

(Note that if a decimal escape sequence is to be followed by a digit,  
it must be expressed using exactly three digits.)

The `@x{UTF-8}` encoding  
of a `@x{Unicode}` character  
can be inserted in a literal string with  
the escape sequence `@T{u{@rep{XXX}}}`  
(with mandatory enclosing braces),

where @rep{XXX} is a sequence of one or more hexadecimal digits representing the character code point.

This code point can be any value less than @M{2@sp{31}}.

(Lua uses the original UTF-8 specification here, which is not restricted to valid Unicode code points.)

Literal strings can also be defined using a long format enclosed by @def{long brackets}.

We define an @def{opening long bracket of level @rep{n}} as an opening square bracket followed by @rep{n} equal signs followed by another opening square bracket.

So, an opening long bracket of @N{level 0} is written as @T{[[, @C{]]}

an opening long bracket of @N{level 1} is written as @T{[=[, @C{]]}

and so on.

A @emph{closing long bracket} is defined similarly;

for instance,

a closing long bracket of @N{level 4} is written as @C{[[ @T{]===}]}

A @def{long literal} starts with an opening long bracket of any level and

ends at the first closing long bracket of the same level.

It can contain any text except a closing bracket of the same level.

Literals in this bracketed form can run for several lines,

do not interpret any escape sequences,

and ignore long brackets of any other level.

Any kind of end-of-line sequence

(carriage return, newline, carriage return followed by newline,

or newline followed by carriage return)

is converted to a simple newline.

When the opening long bracket is immediately followed by a newline, the newline is not included in the string.

As an example, in a system using ASCII

(in which @Char{a} is coded @N{as 97},

newline is coded @N{as 10}, and @Char{1} is coded @N{as 49}),

the five literal strings below denote the same string:

```
@verbatim{
a = 'alo\n123"'
a = "alo\n123\""
a = "\97lo\10\04923"
a = [[alo
123"]]
a = [=[
alo
123"]=]
}
```

Any byte in a literal string not

explicitly affected by the previous rules represents itself.

However, Lua opens files for parsing in text mode, and the system's file functions may have problems with some control characters. So, it is safer to represent binary data as a quoted literal with explicit escape sequences for the non-text characters.

A `@def{numeric constant}` (or `@def{numeral}`) can be written with an optional fractional part and an optional decimal exponent, marked by a letter `@Char{e}` or `@Char{E}`. Lua also accepts `@x{hexadecimal constants}`, which start with `@T{0x}` or `@T{0X}`. Hexadecimal constants also accept an optional fractional part plus an optional binary exponent, marked by a letter `@Char{p}` or `@Char{P}` and written in decimal. (For instance, `@T{0x1.fp10}` denotes 1984, which is `@M{0x1f / 16}` multiplied by `@M{2@sp{10}}`.)

A numeric constant with a radix point or an exponent denotes a float; otherwise, if its value fits in an integer or it is a hexadecimal constant, it denotes an integer; otherwise (that is, a decimal integer numeral that overflows), it denotes a float. Hexadecimal numerals with neither a radix point nor an exponent always denote an integer value; if the value overflows, it `@emph{wraps around}` to fit into a valid integer.

Examples of valid integer constants are

```
@verbatim{
3 345 0xff 0xBEBADA
}
```

Examples of valid float constants are

```
@verbatim{
3.0 3.1416 314.16e-2 0.31416E1 34e1
0x0.1E 0xA23p-4 0X1.921FB54442D18P+1
}
```

A `@def{comment}` starts with a double hyphen (`@T{--}`) anywhere outside a string. If the text immediately after `@T{--}` is not an opening long bracket, the comment is a `@def{short comment}`, which runs until the end of the line.

Otherwise, it is a `@def{long comment}`,  
which runs until the corresponding closing long bracket.

}

`@sect2{variables| @title{Variables}}`

Variables are places that store values.

There are three kinds of variables in Lua:

global variables, local variables, and table fields.

A single name can denote a global variable or a local variable

(or a function's formal parameter,

which is a particular kind of

local variable):

`@Produc{`

`@producname{var}@producbody{@bnfNter{Name}}`

}

`@bnfNter{Name}` denotes identifiers `@see{lexical}`.

Any variable name is assumed to be global unless explicitly declared

as a local `@see{localvar}`.

`@x{Local variables}` are `@emph{lexically scoped}`:

local variables can be freely accessed by functions

defined inside their scope `@see{visibility}`.

Before the first assignment to a variable, its value is `@nil`.

Square brackets are used to index a table:

`@Produc{`

`@producname{var}@producbody{prefixexp @bnfter{[] exp @bnfter{[]}}`

}

The meaning of accesses to table fields can be changed via metatables

`@see{metatable}`.

The syntax `@id{var.Name}` is just syntactic sugar for

`@T{var["Name"]}`:

`@Produc{`

`@producname{var}@producbody{prefixexp @bnfter{.} @bnfNter{Name}}`

}

An access to a global variable `@id{x}`

is equivalent to `@id{_ENV.x}`.

Due to the way that chunks are compiled,

the variable `@id{_ENV}` itself is never global `@see{globalenv}`.

}

@sect2{stats| @title{Statements}

@simplesect{

Lua

supports an almost conventional set of @x{statements},  
similar to those in other conventional languages.

This set includes

blocks, assignments, control structures, function calls,  
and variable declarations.

}

@sect3{@title{Blocks}

A @x{block} is a list of statements,  
which are executed sequentially:

@Produc{

@producname{block}@producbody{@bnfrep{stat}}

}

Lua has @def{empty statements}

that allow you to separate statements with semicolons,  
start a block with a semicolon

or write two semicolons in sequence:

@Produc{

@producname{stat}@producbody{@bnfter{;}}

}

Both function calls and assignments

can start with an open parenthesis.

This possibility leads to an ambiguity in Lua's grammar.

Consider the following fragment:

@verbatim{

a = b + c

(print or io.write)('done')

}

The grammar could see this fragment in two ways:

@verbatim{

a = b + c(print or io.write)('done')

a = b + c; (print or io.write)('done')

}

The current parser always sees such constructions

in the first way,

interpreting

the open parenthesis

as the start of the arguments to a call.

To avoid this ambiguity,

it is a good practice to always precede with a semicolon  
statements that start with a parenthesis:

```
@verbatim{
;(print or io.write)('done')
}
```

A block can be explicitly delimited to produce a single statement:

```
@Produc{
@producname{stat}@producbody{@Rw{do} block @Rw{end}}
}
```

Explicit blocks are useful  
to control the scope of variable declarations.  
Explicit blocks are also sometimes used to  
add a `@Rw{return}` statement in the middle  
of another block `@see{control}`.

```
}
```

```
@sect3{chunks| @title{Chunks}
```

The unit of compilation of Lua is called a `@def{chunk}`.  
Syntactically,  
a chunk is simply a block:

```
@Produc{
@producname{chunk}@producbody{block}
}
```

Lua handles a chunk as the body of an anonymous function  
with a variable number of arguments  
`@see{func-def}`.

As such, chunks can define local variables,  
receive arguments, and return values.

Moreover, such anonymous function is compiled as in the  
scope of

an external local variable called `@id{_ENV}` `@see{globalenv}`.

The resulting function always has `@id{_ENV}` as its only external variable,  
even if it does not use that variable.

A chunk can be stored in a file or in a string inside the host program.

To execute a chunk,

Lua first *@emph{loads}* it,

precompiling the chunk's code into instructions for a virtual machine,  
and then Lua executes the compiled code  
with an interpreter for the virtual machine.

Chunks can also be precompiled into binary form;

see the program `@idx{luac}` and the function `@Lid{string.dump}` for details.

Programs in source and compiled forms are interchangeable;



Lua automatically detects the file type and acts accordingly @seeF{load}.

}

@sect3{assignment| @title{Assignment}}

Lua allows @x{multiple assignments}.

Therefore, the syntax for assignment defines a list of variables on the left side and a list of expressions on the right side.

The elements in both lists are separated by commas:

```
@Produc{
@producname{stat}@producbody{varlist
@bnfter{=} explist}
@producname{varlist}@producbody{var @bnfrep{@bnfter{,} var}}
@producname{explist}@producbody{exp @bnfrep{@bnfter{,} exp}}
}
```

Expressions are discussed in @See{expressions}.

Before the assignment,  
the list of values is @emph{adjusted} to the length of  
the list of variables @see{multires}.

If a variable is both assigned and read  
inside a multiple assignment,  
Lua ensures that all reads get the value of the variable  
before the assignment.

Thus the code

```
@verbatim{
i = 3
i, a[i] = i+1, 20
}
sets @T{a[3]} to 20, without affecting @T{a[4]}
because the @id{i} in @T{a[i]} is evaluated (to 3)
before it is @N{assigned 4}.
```

Similarly, the line

```
@verbatim{
x, y = y, x
}
exchanges the values of @id{x} and @id{y},
and
@verbatim{
x, y, z = y, z, x
}
cyclically permutes the values of @id{x}, @id{y}, and @id{z}.
```

Note that this guarantee covers only accesses  
syntactically inside the assignment statement.

If a function or a metamethod called during the assignment changes the value of a variable, Lua gives no guarantees about the order of that access.

An assignment to a global name `@T{x = val}` is equivalent to the assignment `@T{_ENV.x = val} @see{globalenv}`.

The meaning of assignments to table fields and global variables (which are actually table fields, too) can be changed via metatables `@see{metatable}`.

}

`@sect3{control| @title{Control Structures}`

The control structures

`@Rw{if}`, `@Rw{while}`, and `@Rw{repeat}` have the usual meaning and familiar syntax:

`@index{while-do statement}`

`@index{repeat-until statement}`

`@index{if-then-else statement}`

`@Produc{`

`@producname{stat}@produciblebody{@Rw{while} exp @Rw{do} block @Rw{end}}`

`@producname{stat}@produciblebody{@Rw{repeat} block @Rw{until} exp}`

`@producname{stat}@produciblebody{@Rw{if} exp @Rw{then} block`

`@bnfrep{@Rw{elseif} exp @Rw{then} block}`

`@bnfoft{@Rw{else} block} @Rw{end}}`

}

Lua also has a `@Rw{for}` statement, in two flavors `@see{for}`.

The `@x{condition expression}` of a control structure can return any value.

Both

`@false` and `@nil` test false.

All values different from `@nil` and `@false` test true.

In particular, the number 0 and the empty string also test true.

In the `@Rw{repeat}@En@Rw{until}` loop, the inner block does not end at the `@Rw{until}` keyword, but only after the condition.

So, the condition can refer to local variables declared inside the loop block.

The `@Rw{goto}` statement transfers the program control to a label.

For syntactical reasons,

labels in Lua are considered statements too:

`@index{goto statement}`

```

@index{label}
@Produc{
@producname{stat}@producbody{@Rw{goto} Name}
@producname{stat}@producbody{label}
@producname{label}@producbody{@bnfter{::} Name @bnfter{::}}
}

```

A label is visible in the entire block where it is defined, except inside nested functions.

A goto can jump to any visible label as long as it does not enter into the scope of a local variable.

A label should not be declared where a label with the same name is visible, even if this other label has been declared in an enclosing block.

The @Rw{break} statement terminates the execution of a @Rw{while}, @Rw{repeat}, or @Rw{for} loop, skipping to the next statement after the loop:

```

@index{break statement}
@Produc{
@producname{stat}@producbody{@Rw{break}}
}

```

A @Rw{break} ends the innermost enclosing loop.

The @Rw{return} statement is used to return values from a function or a chunk (which is handled as an anonymous function).

```
@index{return statement}
```

Functions can return more than one value, so the syntax for the @Rw{return} statement is

```

@Produc{
@producname{stat}@producbody{@Rw{return} @bnfopt{explist} @bnfopt{@bnfter{;}}}
}

```

The @Rw{return} statement can only be written as the last statement of a block.

If it is necessary to @Rw{return} in the middle of a block, then an explicit inner block can be used, as in the idiom @T{do return end},

because now @Rw{return} is the last statement in its (inner) block.

```
}
```

```
@sect3{for| @title{For Statement}
```

```
@index{for statement}
```

The @Rw{for} statement has two forms:

one

numerical and one generic.

```
@sect4{@title{The numerical @Rw{for} loop}
```

The numerical `@Rw{for}` loop repeats a block of code while a control variable goes through an arithmetic progression.

It has the following syntax:

```
@Produc{  
@producname{stat}@producbody{@Rw{for} @bnfNter{Name} @bnfter{=}  
exp @bnfter{,} exp @bnfopt{@bnfter{,} exp} @Rw{do} block @Rw{end}}  
}
```

The given identifier (`@bnfNter{Name}`) defines the control variable, which is a new variable local to the loop body (`@emph{block}`).

The loop starts by evaluating once the three control expressions.

Their values are called respectively

the `@emph{initial value}`, the `@emph{limit}`, and the `@emph{step}`.

If the step is absent, it defaults `@N{to 1}`.

If both the initial value and the step are integers,

the loop is done with integers;

note that the limit may not be an integer.

Otherwise, the three values are converted to

floats and the loop is done with floats.

Beware of floating-point accuracy in this case.

After that initialization,

the

loop body is repeated with the value of the control variable

going through an arithmetic progression,

starting at the initial value,

with a common difference given by the step.

A negative step makes a decreasing sequence;

a step equal to zero raises an error.

The loop continues while the value is less than

or equal to the limit

(greater than or equal to for a negative step).

If the initial value is already greater than the limit

(or less than, if the step is negative),

the body is not executed.

For integer loops,

the control variable never wraps around;

instead, the loop ends in case of an overflow.

You should not change the value of the control variable

during the loop.

If you need its value after the loop,  
assign it to another variable before exiting the loop.

```
}
```

```
@sect4{@title{The generic @Rw{for} loop}
```

The generic @Rw{for} statement works over functions,  
called @def{iterators}.

On each iteration, the iterator function is called to produce a new value,  
stopping when this new  
value is @nil.

The generic @Rw{for} loop has the following syntax:

```
@Produc{  
@producname{stat}@produciblebody{@Rw{for} namelist @Rw{in} explist  
    @Rw{do} block @Rw{end}}  
@producname{namelist}@produciblebody{@bnfNter{Name} @bnfrep{@bnfter{,} @bnfNter{Name}}}  
}
```

A @Rw{for} statement like

```
@verbatim{  
for @rep{var_1}, @Cdots, @rep{var_n} in @rep{explist} do @rep{body} end  
}
```

works as follows.

The names @rep{var\_i} declare loop variables local to the loop body.  
The first of these variables is the @emph{control variable}.

The loop starts by evaluating @rep{explist}  
to produce four values:  
an @emph{iterator function},  
a @emph{state},  
an initial value for the control variable,  
and a @emph{closing value}.

Then, at each iteration,  
Lua calls the iterator function with two arguments:  
the state and the control variable.  
The results from this call are then assigned to the loop variables,  
following the rules of multiple assignments @see{assignment}.  
If the control variable becomes @nil,  
the  
loop terminates.  
Otherwise, the body is executed and the loop goes  
to the next iteration.

The closing value behaves like a  
to-be-closed variable @see{to-be-closed},

which can be used to release resources when the loop ends.  
Otherwise, it does not interfere with the loop.

You should not change the value of the control variable during the loop.

```
}
```

```
}
```

@sect3{funcstat| @title{Function Calls as Statements}}

To allow possible side-effects,  
function calls can be executed as statements:

```
@Produc{  
@producname{stat}@producbody{functioncall}  
}
```

In this case, all returned values are thrown away.  
Function calls are explained in @See{functioncall}.

```
}
```

@sect3{localvar| @title{Local Declarations}}

@x{Local variables} can be declared anywhere inside a block.  
The declaration can include an initialization:

```
@Produc{  
@producname{stat}@producbody{@Rw{local} attnamelist @bnfopt{@bnfter{=} explist}}  
@producname{attnamelist}@producbody{  
@bnfNter{Name} attrib @bnfrep{@bnfter{,} @bnfNter{Name} attrib}}  
}
```

If  
present, an initial assignment has the same semantics  
of a multiple assignment @see{assignment}.  
Otherwise, all variables are initialized with @nil.

Each variable name may be postfixed by an attribute  
(a name between angle brackets):

```
@Produc{  
@producname{attrib}@producbody{@bnfopt{@bnfter{<} @bnfNter{Name} @bnfter{>}}}  
}
```

There are two possible attributes:

@id{const}, which declares a @x{constant variable},  
that is, a variable that cannot be assigned to  
after its initialization;  
and @id{close}, which declares a to-be-closed variable @see{to-be-closed}.  
A list of variables can contain at most one to-be-closed variable.

A chunk is also a block @see{chunks},  
and so local variables can be declared in a chunk outside any explicit block.

The visibility rules for local variables are explained in `@See{visibility}`.

}

`@sect3{to-be-closed| @title{To-be-closed Variables}`

A to-be-closed variable behaves like a constant local variable, except that its value is `@emph{closed}` whenever the variable goes

out of scope, including normal block termination, exiting its block by `@Rw{break}/@Rw{goto}/@Rw{return}`, or exiting by an error.

Here, to `@emph{close}` a value means to call its `@idx{__close}` metamethod.

When calling the metamethod, the value itself is passed as the first argument and the error object that caused the exit (if any) is passed as a second argument; if there was no error, the second argument is `@nil`.

The value assigned to a to-be-closed variable must have a `@idx{__close}` metamethod or be a false value.  
(`@nil` and `@false` are ignored as to-be-closed values.)

If several to-be-closed variables go out of scope at the same event, they are closed in the reverse order that they were declared.

If there is any error while running a closing method, that error is handled like an error in the regular code where the variable was defined.  
After an error, the other pending closing methods will still be called.

If a coroutine yields and is never resumed again, some variables may never go out of scope, and therefore they will never be closed.  
(These variables are the ones created inside the coroutine and in scope at the point where the coroutine yielded.)  
Similarly, if a coroutine ends with an error, it does not unwind its stack, so it does not close any variable.  
In both cases, you can either use finalizers or call `@Lid{coroutine.close}` to close the variables.

However, if the coroutine was created through `@Lid{coroutine.wrap}`, then its corresponding function will close the coroutine in case of errors.

```
}
```

```
}
```

```
@sect2{expressions| @title{Expressions}}
```

```
@simplesect{
```

The basic expressions in Lua are the following:

```
@Produc{
@producname{exp}@producbody{prefixexp}
@producname{exp}@producbody{@Rw{nil} @Or @Rw{false} @Or @Rw{true}}
@producname{exp}@producbody{@bnfNter{Numeral}}
@producname{exp}@producbody{@bnfNter{LiteralString}}
@producname{exp}@producbody{functiondef}
@producname{exp}@producbody{tableconstructor}
@producname{exp}@producbody{@bnfter{...}}
@producname{exp}@producbody{exp
binop exp}
@producname{exp}@producbody{unop exp}
@producname{prefixexp}@producbody{var @Or functioncall @Or
@bnfter{() exp @bnfter{}}}
}
```

Numerals and literal strings are explained in `@See{lexical}`;  
variables are explained in `@See{variables}`;  
function definitions are explained in `@See{func-def}`;  
function calls are explained in `@See{functioncall}`;  
table constructors are explained in `@See{tableconstructor}`.  
Vararg expressions,  
denoted by three dots (`@Char{...}`), can only be used when  
directly inside a variadic function;  
they are explained in `@See{func-def}`.

Binary operators comprise arithmetic operators `@see{arith}`,  
bitwise operators `@see{bitwise}`,  
relational operators `@see{rel-ops}`, logical operators `@see{logic}`,  
and the concatenation operator `@see{concat}`.  
Unary operators comprise the unary minus `@see{arith}`,  
the unary bitwise NOT `@see{bitwise}`,  
the unary logical `@Rw{not} @see{logic}`,  
and the unary `@def{length operator} @see{len-op}`.



}

```
@sect3{arith|
  @title{Arithmetic Operators}
  Lua supports the following @x{arithmetic operators}:
  @description{
    @item{@T{+}|addition}
    @item{@T{-}|subtraction}
    @item{@T{*}|multiplication}
    @item{@T{/}|float division}
    @item{@T{//}|floor division}
    @item{@T{%}|modulo}
    @item{@T{^}|exponentiation}
    @item{@T{-}|unary minus}
  }
```

With the exception of exponentiation and float division, the arithmetic operators work as follows:

If both operands are integers, the operation is performed over integers and the result is an integer.

Otherwise, if both operands are numbers, then they are converted to floats, the operation is performed following the machine's rules for floating-point arithmetic (usually the @x{IEEE 754} standard), and the result is a float.

(The string library coerces strings to numbers in arithmetic operations; see @See{coercion} for details.)

Exponentiation and float division (@T{/}) always convert their operands to floats and the result is always a float.

Exponentiation uses the @ANSI{pow}, so that it works for non-integer exponents too.

Floor division (@T{//}) is a division that rounds the quotient towards minus infinity, resulting in the floor of the division of its operands.

Modulo is defined as the remainder of a division that rounds the quotient towards minus infinity (floor division).

In case of overflows in integer arithmetic, all operations @emphx{wrap around}.

}

@sect3{bitwise| @title{Bitwise Operators}

Lua supports the following @x{bitwise operators}:

@description{

@item{@T{&}}|bitwise AND}

@item{@T{@VerBar}}|bitwise OR}

@item{@T{~}}|bitwise exclusive OR}

@item{@T{>>}}|right shift}

@item{@T{<<}}|left shift}

@item{@T{~}}|unary bitwise NOT}

}

All bitwise operations convert its operands to integers

@see{coercion},

operate on all bits of those integers,

and result in an integer.

Both right and left shifts fill the vacant bits with zeros.

Negative displacements shift to the other direction;

displacements with absolute values equal to or higher than the

number of bits in an integer

result in zero (as all bits are shifted out).

}

@sect3{coercion| @title{Coercions and Conversions}

Lua provides some automatic conversions between some types and representations at run time.

Bitwise operators always convert float operands to integers.

Exponentiation and float division

always convert integer operands to floats.

All other arithmetic operations applied to mixed numbers

(integers and floats) convert the integer operand to a float.

The C API also converts both integers to floats and

floats to integers, as needed.

Moreover, string concatenation accepts numbers as arguments, besides strings.

In a conversion from integer to float,

if the integer value has an exact representation as a float, that is the result.

Otherwise,

the conversion gets the nearest higher or

the nearest lower representable value.

This kind of conversion never fails.

The conversion from float to integer checks whether the float has an exact representation as an integer (that is, the float has an integral value and it is in the range of integer representation). If it does, that representation is the result. Otherwise, the conversion fails.

Several places in Lua coerce strings to numbers when necessary. In particular, the string library sets metamethods that try to coerce strings to numbers in all arithmetic operations. If the conversion fails, the library calls the metamethod of the other operand (if present) or it raises an error. Note that bitwise operators do not do this coercion.

It is always a good practice not to rely on the implicit coercions from strings to numbers, as they are not always applied; in particular, `@T{"1"==1}` is false and `@T{"1"<1}` raises an error `@see{rel-ops}`. These coercions exist mainly for compatibility and may be removed in future versions of the language.

A string is converted to an integer or a float following its syntax and the rules of the Lua lexer. The string may have also leading and trailing whitespaces and a sign. All conversions from strings to numbers accept both a dot and the current locale mark as the radix character. (The Lua lexer, however, accepts only a dot.) If the string is not a valid numeral, the conversion fails. If necessary, the result of this first step is then converted to a specific number subtype following the previous rules for conversions between floats and integers.

The conversion from numbers to strings uses a non-specified human-readable format. To convert numbers to strings in any specific way, use the function `@Lid{string.format}`.

}

`@sect3{rel-ops| @title{Relational Operators}`  
Lua supports the following `@x{relational operators}`:

```

@description{
@item{@T{==}|equality}
@item{@T{~=}|inequality}
@item{@T{<}|less than}
@item{@T{>}|greater than}
@item{@T{<=}|less or equal}
@item{@T{>=}|greater or equal}
}

```

These operators always result in `@false` or `@true`.

Equality (`@T{==}`) first compares the type of its operands.

If the types are different, then the result is `@false`.

Otherwise,

the values of the operands are compared.

Strings are equal if they have the same byte content.

Numbers are equal if they denote the same mathematical value.

Tables, userdata, and threads

are compared by reference:

two objects are considered equal only if they are the same object.

Every time you create a new object

(a table, a userdata, or a thread),

this new object is different from any previously existing object.

A function is always equal to itself.

Functions with any detectable difference

(different behavior, different definition) are always different.

Functions created at different times but with no detectable differences

may be classified as equal or not

(depending on internal caching details).

You can change the way that Lua compares tables and userdata

by using the `@idx{__eq}` metamethod `@see{metatable}`.

Equality comparisons do not convert strings to numbers

or vice versa.

Thus, `@T{"0"]==0}` evaluates to `@false`,

and `@T{t[0]}` and `@T{t["0"]}` denote different

entries in a table.

The

operator `@T{~=}` is exactly the negation of equality (`@T{==}`).

The order operators work as follows.

If both arguments are numbers,

then they are compared according to their mathematical values,

regardless of their subtypes.

Otherwise, if both arguments are strings,

then their values are compared according to the current locale.

Otherwise, Lua tries to call the `@idx{__lt}` or the `@idx{__le}` metamethod `@see{metatable}`.

A comparison `@T{a > b}` is translated to `@T{b < a}` and `@T{a >= b}` is translated to `@T{b <= a}`.

Following the `@x{IEEE 754}` standard, the special value `@x{NaN}` is considered neither less than, nor equal to, nor greater than any value, including itself.

}

`@sect3{logic| @title{Logical Operators}`

The `@x{logical operators}` in Lua are

`@Rw{and}`, `@Rw{or}`, and `@Rw{not}`.

Like the control structures `@see{control}`, all logical operators consider both `@false` and `@nil` as false and anything else as true.

The negation operator `@Rw{not}` always returns `@false` or `@true`.

The conjunction

operator `@Rw{and}` returns its first argument

if this value is `@false` or `@nil`;

otherwise, `@Rw{and}` returns its second argument.

The disjunction operator `@Rw{or}` returns its first argument

if this value is different from `@nil` and `@false`;

otherwise, `@Rw{or}` returns its second argument.

Both `@Rw{and}` and `@Rw{or}` use `@x{short-circuit evaluation}`;

that is,

the second operand is evaluated only if necessary.

Here are some examples:

`@verbatim{`

`10 or 20       --> 10`

`10 or error()  --> 10`

`nil or "a"     --> "a"`

`nil and 10     --> nil`

`false and error() --> false`

`false and nil   --> false`

`false or nil    --> nil`

`10 and 20       --> 20`

`}`

}

`@sect3{concat| @title{Concatenation}`

The string `@x{concatenation}` operator in Lua is denoted by two dots (`@Char{..}`).

If both operands are strings or numbers, then the numbers are converted to strings

in a non-specified format @see{coercion}.

Otherwise, the @idx{\_\_concat} metamethod is called @see{metatable}.

}

@sect3{len-op|  
@title{The Length Operator}}

The length operator is denoted by the unary prefix operator @T{#}.

The length of a string is its number of bytes.  
(That is the usual meaning of string length when each  
character is one byte.)

The length operator applied on a table  
returns a @x{border} in that table.  
A @def{border} in a table @id{t} is any non-negative integer  
that satisfies the following condition:

@verbatim{  
(border == 0 or t[border] ~= nil) and  
(t[border + 1] == nil or border == math.maxinteger)  
}

In words,  
a border is any positive integer index present in the table  
that is followed by an absent index,  
plus two limit cases:  
zero, when index 1 is absent;  
and the maximum value for an integer, when that index is present.  
Note that keys that are not positive integers  
do not interfere with borders.

A table with exactly one border is called a @def{sequence}.

For instance, the table @T{{10, 20, 30, 40, 50}} is a sequence,  
as it has only one border (5).

The table @T{{10, 20, 30, nil, 50}} has  
two borders (3 and 5),  
and therefore it is not a sequence.  
(The @nil at index 4 is called a @emphx{hole}.)

The table @T{{nil, 20, 30, nil, nil, 60, nil}}  
has three borders (0, 3, and 6),  
so it is not a sequence, too.

The table @T{{}} is a sequence with border 0.

When @id{t} is a sequence,  
@T{#t} returns its only border,  
which corresponds to the intuitive notion of the length of the sequence.  
When @id{t} is not a sequence,  
@T{#t} can return any of its borders.

(The exact one depends on details of the internal representation of the table, which in turn can depend on how the table was populated and the memory addresses of its non-numeric keys.)

The computation of the length of a table has a guaranteed worst time of  $\mathcal{O}(\log n)$ , where  $M\{n\}$  is the largest integer key in the table.

A program can modify the behavior of the length operator for any value but strings through the `@idx{__len}` metamethod `@see{metatable}`.

}

`@sect3{prec| @title{Precedence}`  
`@x{Operator precedence}` in Lua follows the table below, from lower to higher priority:

`@verbatim{`  
or  
and  
`< > <= >= ~= ==`  
`|`  
`~`  
`&`  
`<< >>`  
`..`  
`+ -`  
`* / // %`  
unary operators (`not # - ~`)  
`^`  
`}`

As usual, you can use parentheses to change the precedences of an expression. The concatenation (`@Char{..}`) and exponentiation (`@Char{^}`) operators are right associative. All other binary operators are left associative.

}

`@sect3{tableconstructor| @title{Table Constructors}`  
Table `@x{constructors}` are expressions that create tables. Every time a constructor is evaluated, a new table is created. A constructor can be used to create an empty table or to create a table and initialize some of its fields. The general syntax for constructors is  
`@Produc{`  
`@producname{tableconstructor}@producbody{@bnfter{@Open} @bnfopt{fieldlist} @bnfter{@Close}}`

```

@producname{fieldlist}@producbody{field @bnfrep{fieldsep field} @bnfopt{fieldsep}}
@producname{field}@producbody{@bnfter{[] exp @bnfter{[]} @bnfter{=} exp @Or

    @bnfNter{Name} @bnfter{=} exp @Or exp}
@producname{fieldsep}@producbody{@bnfter{,} @Or @bnfter{;}}
}

```

Each field of the form `@T{[exp1] = exp2}` adds to the new table an entry with key `@id{exp1}` and value `@id{exp2}`.

A field of the form `@T{name = exp}` is equivalent to `@T{["name"] = exp}`.

Fields of the form `@id{exp}` are equivalent to `@T{[i] = exp}`, where `@id{i}` are consecutive integers starting with 1; fields in the other formats do not affect this counting.

For example,

```

@verbatim{
a = { [f(1)] = g; "x", "y"; x = 1, f(x), [30] = 23; 45 }
}

```

is equivalent to

```

@verbatim{
do
  local t = {}
  t[f(1)] = g
  t[1] = "x"      -- 1st exp
  t[2] = "y"      -- 2nd exp
  t.x = 1         -- t["x"] = 1
  t[3] = f(x)     -- 3rd exp
  t[30] = 23
  t[4] = 45       -- 4th exp
  a = t
end
}

```

The order of the assignments in a constructor is undefined.  
(This order would be relevant only when there are repeated keys.)

If the last field in the list has the form `@id{exp}` and the expression is a multires expression, then all values returned by this expression enter the list consecutively `@see{multires}`.

The field list can have an optional trailing separator, as a convenience for machine-generated code.

```

}

```



@sect3{functioncall| @title{Function Calls}

A @x{function call} in Lua has the following syntax:

```
@Produc{
@producname{functioncall}@producbody{prefixexp args}
}
```

In a function call,

first @bnfNter{prefixexp} and @bnfNter{args} are evaluated.

If the value of @bnfNter{prefixexp} has type @emph{function},  
then this function is called

with the given arguments.

Otherwise, if present,

the @bnfNter{prefixexp} @idx{\_\_call} metamethod is called:

its first argument is the value of @bnfNter{prefixexp},

followed by the original call arguments

@see{metatable}.

The form

```
@Produc{
@producname{functioncall}@producbody{prefixexp @bnfter{:} @bnfNter{Name} args}
}
```

can be used to emulate methods.

A call @T{v:name(@rep{args})}

is syntactic sugar for @T{v.name(v,@rep{args})},

except that @id{v}

is evaluated only once.

Arguments have the following syntax:

```
@Produc{
@producname{args}@producbody{@bnfter{({ @bnfopt{explist} @bnfter{}})}
@producname{args}@producbody{tableconstructor}
@producname{args}@producbody{@bnfNter{LiteralString}}
}
```

All argument expressions are evaluated before the call.

A call of the form @T{f{@rep{fields}}} is

syntactic sugar for @T{f({@rep{fields}})};

that is, the argument list is a single new table.

A call of the form @T{f'@rep{string}'}

(or @T{f"@rep{string}" or @T{f[[@rep{string}]])

is syntactic sugar for @T{f('@rep{string}')};

that is, the argument list is a single literal string.

A call of the form @T{return @rep{functioncall}} not in the  
scope of a to-be-closed variable is called a @def{tail call}.

Lua implements @def{proper tail calls}

(or @def{proper tail recursion}):

In a tail call,

the called function reuses the stack entry of the calling function.

Therefore, there is no limit on the number of nested tail calls that

a program can execute.

However,

a tail call erases any debug information about the calling function.

Note that a tail call only happens with a particular syntax, where the `@Rw{return}` has one single function call as argument, and it is outside the scope of any to-be-closed variable.

This syntax makes the calling function return exactly the returns of the called function, without any intervening action.

So, none of the following examples are tail calls:

```
@verbatim{
return (f(x))    -- results adjusted to 1
return 2 * f(x)  -- result multiplied by 2
return x, f(x)   -- additional results
f(x); return    -- results discarded
return x or f(x) -- results adjusted to 1
}

}
```

```
@sect3{func-def| @title{Function Definitions}}
```

The syntax for function definition is

```
@Produc{
@producname{functiondef}@producbody{@Rw{function} funcbody}
@producname{funcbody}@producbody{@bnfter{() @bnfopt{parlist} @bnfter{}} block @Rw{end}}
}
```

The following syntactic sugar simplifies function definitions:

```
@Produc{
@producname{stat}@producbody{@Rw{function}
funcname funcbody}
@producname{stat}@producbody{@Rw{local} @Rw{function} @bnfNter{Name} funcbody}
@producname{funcname}@producbody{@bnfNter{Name} @bnfrep{@bnfter{.} @bnfNter{Name}}
@bnfopt{@bnfter{:} @bnfNter{Name}}}
}
```

The statement

```
@verbatim{
function f () @rep{body} end
}
```

translates to

```
@verbatim{
f = function () @rep{body} end
}
```

The statement

```
@verbatim{
function t.a.b.c.f () @rep{body} end
```

```
}
```

translates to

```
@verbatim{
```

```
t.a.b.c.f = function () @rep{body} end
```

```
}
```

The statement

```
@verbatim{
```

```
local function f () @rep{body} end
```

```
}
```

translates to

```
@verbatim{
```

```
local f; f = function () @rep{body} end
```

```
}
```

not to

```
@verbatim{
```

```
local f = function () @rep{body} end
```

```
}
```

(This only makes a difference when the body of the function contains references to `@id{f}`.)

A function definition is an executable expression, whose value has type `@emph{function}`.

When Lua precompiles a chunk,

all its function bodies are precompiled too,

but they are not created yet.

Then, whenever Lua executes the

function definition,

the function is `@emph{instantiated}` (or `@emph{closed}`).

This function instance, or `@emphx{closure}`,

is the final value of the expression.

Parameters act as local variables that are initialized with the argument values:

```
@Produc{
```

```
@producname{parlist}@produciblebody{namelist @bnfopt{@bnfter{,} @bnfter{...}} @Or
```

```
@bnfter{...}}
```

```
}
```

When a Lua function is called,

it adjusts its list of `@x{arguments}` to

the length of its list of parameters `@see{multires}`,

unless the function is a `@def{variadic function}`,

which is indicated by three dots (`@Char{...}`)

at the end of its parameter list.

A variadic function does not adjust its argument list;

instead, it collects all extra arguments and supplies them

to the function through a `@def{vararg expression}`,

which is also written as three dots.

The value of this expression is a list of all actual extra arguments,

similar to a function with multiple results @see{multires}.

As an example, consider the following definitions:

```
@verbatim{
function
  f(a, b) end
function g(a, b, ...) end
function r() return 1,2,3 end
}
```

Then, we have the following mapping from arguments to parameters and to the vararg expression:

```
@verbatim{
CALL      PARAMETERS
```

```

f(3)      a=3, b=nil
f(3, 4)   a=3, b=4
f(3, 4, 5) a=3, b=4
f(r(), 10) a=1, b=10
f(r())    a=1, b=2
```

```

g(3)      a=3, b=nil, ... --> (nothing)
g(3, 4)   a=3, b=4,  ... --> (nothing)
g(3, 4, 5, 8) a=3, b=4,  ... --> 5 8
g(5, r())  a=5, b=1,  ... --> 2 3
}
```

Results are returned using the @Rw{return} statement @see{control}.

If control reaches the end of a function  
without encountering a @Rw{return} statement,  
then the function returns with no results.

@index{multiple return}

There is a system-dependent limit on the number of values  
that a function may return.

This limit is guaranteed to be greater than 1000.

The @emphx{colon} syntax  
is used to emulate @def{methods},  
adding an implicit extra parameter @idx{self} to  
the function.

Thus, the statement

```
@verbatim{
function t.a.b.c:f (@rep{params}) @rep{body} end
}
```

is syntactic sugar for

```
@verbatim{
t.a.b.c.f = function (self, @rep{params}) @rep{body} end
```

}

}

@sect3{multires| @title{Lists of expressions, multiple results,  
and adjustment}

Both function calls and vararg expressions can result in multiple values.  
These expressions are called @def{multires expressions}.

When a multires expression is used as the last element  
of a list of expressions,  
all results from the expression are added to the  
list of values produced by the list of expressions.  
Note that a single expression  
in a place that expects a list of expressions  
is the last expression in that (singleton) list.

These are the places where Lua expects a list of expressions:

@description{

@item{A @rw{return} statement,  
for instance @T{return e1, e2, e3} @see{control}.}

@item{A table constructor,  
for instance @T{{e1, e2, e3}} @see{tableconstructor}.}

@item{The arguments of a function call,  
for  
instance @T{foo(e1, e2, e3)} @see{functioncall}.}

@item{A multiple assignment,  
for instance @T{a , b, c = e1, e2, e3} @see{assignment}.}

@item{A local declaration,  
for instance @T{local a , b, c = e1, e2, e3} @see{localvar}.}

@item{The initial values in a generic @rw{for} loop,  
for instance @T{for k in e1, e2, e3 do ... end} @see{for}.}

}

In the last four cases,  
the list of values from the list of expressions  
must be @emph{adjusted} to a specific length:  
the number of parameters in a call to a non-variadic function  
@see{func-def},  
the number of variables in a multiple assignment or  
a local declaration,

and exactly four values for a generic `@rw{for}` loop.

The `@def{adjustment}` follows these rules:

If there are more values than needed,  
the extra values are thrown away;  
if there are fewer values than needed,  
the list is extended with `@nil`'s.

When the list of expressions ends with a multi-res expression,  
all results from that expression enter the list of values  
before the adjustment.

When

a multi-res expression is used  
in a list of expressions without being the last element,  
or in a place where the syntax expects a single expression,  
Lua adjusts the result list of that expression to one element.  
As a particular case,  
the syntax expects a single expression inside a parenthesized expression;  
therefore, adding parentheses around a multi-res expression  
forces it to produce exactly one result.

We seldom need to use a vararg expression in a place  
where the syntax expects a single expression.  
(Usually it is simpler to add a regular parameter before  
the variadic part and use that parameter.)

When there is such a need,  
we recommend assigning the vararg expression  
to a single variable and using that variable  
in its place.

Here are some examples of uses of multi-res expressions.

In all cases, when the construction needs

`@Q{the n-th result}` and there is no such result,  
it uses a `@nil`.

`@verbatim{`

`print(x, f())` -- prints x and all results from f().

`print(x, (f()))` -- prints

x and the first result from f().

`print(f(), x)` -- prints the first result from f() and x.

`print(1 + f())` -- prints 1 added to the first result from f().

`local x = ...` -- x gets the first vararg argument.

`x,y = ...` -- x gets the first vararg argument,  
-- y gets the second vararg argument.

`x,y,z = w, f()` -- x gets w, y gets the first result from f(),  
-- z gets the second result from f().

`x,y,z = f()` -- x gets the first result from f(),  
-- y gets the second result from f(),  
-- z gets the third result from f().

`x,y,z = f(), g()` -- x gets the first result from f(),

```

        -- y gets the first result from g(),
        -- z gets the second result from g().
x,y,z = (f())    -- x gets the first result from f(), y and z get nil.
return f()      -- returns all results from f().
return x, ...   -- returns x and all received vararg arguments.
return x,y,f()
    -- returns x, y, and all results from f().
{f()}          -- creates a list with all results from f().
{...}          -- creates a list with all vararg arguments.
{f(), 5}       -- creates a list with the first result from f() and 5.
}

}

}

```

@sect2{visibility| @title{Visibility Rules}

@index{visibility}

Lua is a lexically scoped language.

The scope of a local variable begins at the first statement after its declaration and lasts until the last non-void statement of the innermost block that includes the declaration.

(@emph{Void statements} are labels and empty statements.)

Consider the following example:

```

@verbatim{
x = 10          -- global variable
do             -- new block
  local x = x   -- new 'x', with value 10
  print(x)     --> 10
  x = x+1
do             -- another block
  local x = x+1 -- another 'x'
  print(x)     --> 12
end
print(x)       --> 11
end
print(x)       --> 10 (the global one)
}

```

Notice

that, in a declaration like @T{local x = x}, the new @id{x} being declared is not in scope yet, and so the second @id{x} refers to the outside variable.

Because of the @x{lexical scoping} rules, local variables can be freely accessed by functions defined inside their scope.

A local variable used by an inner function is called an `@def{upvalue}` (or `@emphx{external local variable}`, or simply `@emphx{external variable}`) inside the inner function.

Notice that each execution of a `@Rw{local}` statement defines new local variables.

Consider the following example:

```
@verbatim{
a = {}
local x = 20
for i = 1, 10 do
  local y = 0
  a[i] = function () y = y + 1; return x + y end
end
}
```

The loop creates ten closures

(that is, ten instances of the anonymous function).

Each of these closures uses a different `@id{y}` variable, while all of them share the same `@id{x}`.

```
}
```

```
}
```

```
@C{-----}
@sect1{API| @title{The Application Program Interface}}
```

```
@simplesect{
```

```
@index{C
API}
```

This section describes the `@N{C API}` for Lua, that is, the set of `@N{C functions}` available to the host program to communicate with Lua.

All API functions and related types and constants are declared in the header file `@defid{lua.h}`.

Even when we use the term `@Q{function}`, any facility in the API may be provided as a macro instead. Except where stated otherwise, all such macros use each of their arguments exactly once (except for the first argument, which is always a Lua state), and so do not generate any hidden side-effects.

As in most `@N{C libraries}`, the Lua API functions do not check their arguments for validity or consistency.



However, you can change this behavior by compiling Lua with the macro `@defid{LUA_USE_APICHECK}` defined.

The Lua library is fully reentrant:  
it has no global variables.  
It keeps all information it needs in a dynamic structure,  
called the `@def{Lua state}`.

Each Lua state has one or more threads,  
which correspond to independent,  
cooperative lines of execution.  
The type `@Lid{lua_State}` (despite its name) refers to a thread.  
(Indirectly, through the thread, it also refers to the  
Lua state associated to the thread.)

A pointer to a thread must be passed as the first argument to  
every function in the library, except to `@Lid{lua_newstate}`,  
which creates a Lua state from scratch and returns a pointer  
to the `@emph{main thread}` in the new state.

}

`@sect2{@title{The Stack}`

`@simplesect{`

Lua uses a `@emph{virtual stack}` to pass values to and from C.  
Each element in this stack represents a Lua value  
(`@nil`, number, string, etc.).  
Functions in the API can access this stack through the  
Lua state parameter that they receive.

Whenever Lua calls C, the called function gets a new stack,  
which is independent of previous stacks and of stacks of  
`@N{C functions}` that are still active.  
This stack initially contains any arguments to the `@N{C function}`  
and it is where the `@N{C function}` can store temporary  
Lua values and must push  
its results  
to be returned to the caller `@seeC{lua_CFunction}`.

For convenience,  
most query operations in the API do not follow a strict stack discipline.  
Instead, they can refer to any element in the stack  
by using an `@emph{index}:@index{index (API stack)}`  
A positive index represents an absolute stack position,  
starting `@N{at 1}` as the bottom of the stack;

a negative index represents an offset relative to the top of the stack.  
More specifically, if the stack has `@rep{n}` elements,  
then `@N{index 1}` represents the first element  
(that is, the element that was pushed onto the stack first)  
and  
`@N{index @rep{n}}` represents the last element;  
`@N{index @num{-1}}` also represents the last element  
(that is, the element at `@N{the top}`)  
and index `@M{-n}` represents the first element.

}

@sect3{stacksize| @title{Stack Size}

When you interact with the Lua API,  
you are responsible for ensuring consistency.  
In particular,  
`@emph{you are responsible for controlling stack overflow}`.  
When you call any  
API function,  
you must ensure the stack has enough room to accommodate the results.

There is one exception to the above rule:  
When you call a Lua function  
without a fixed number of results `@seeF{lua_call}`,  
Lua ensures that the stack has enough space for all results.  
However, it does not ensure any extra space.  
So, before pushing anything on the stack after such a call  
you should use `@Lid{lua_checkstack}`.

Whenever Lua calls C,  
it ensures that the stack has space for  
at least `@defid{LUA_MINSTACK}` extra elements;  
that is, you can safely push up to `@id{LUA_MINSTACK}` values into it.  
`@id{LUA_MINSTACK}` is defined as 20,  
so that usually you do not have to worry about stack space  
unless your code has loops pushing elements onto the stack.  
Whenever necessary,  
you can use the function `@Lid{lua_checkstack}`  
to ensure that the stack has enough space for pushing new elements.

}

@sect3{@title{Valid and Acceptable Indices}

Any function in the API that receives stack indices  
works only with `@emphx{valid`

indices} or @emphx{acceptable indices}.

A @def{valid index} is an index that refers to a position that stores a modifiable Lua value. It comprises stack indices @N{between 1} and the stack top (@T{1 @leq abs(index) @leq top}) @index{stack index} plus @def{pseudo-indices}, which represent some positions that are accessible to @N{C code} but that are not in the stack. Pseudo-indices are used to access the registry @see{registry} and the upvalues of a @N{C function} @see{c-closure}.

Functions that do not need a specific mutable position, but only a value (e.g., query functions), can be called with acceptable indices. An @def{acceptable index} can be any valid index, but it also can be any positive index after the stack top within the space allocated for the stack, that is, indices up to the stack size. (Note that 0 is never an acceptable index.) Indices to upvalues @see{c-closure} greater than the real number of upvalues in the current @N{C function} are also acceptable (but invalid). Except when noted otherwise, functions in the API work with acceptable indices.

Acceptable indices serve to avoid extra tests against the stack top when querying the stack. For instance, a @N{C function} can query its third argument without the need to check whether there is a third argument, that is, without the need to check whether 3 is a valid index.

For functions that can be called with acceptable indices, any non-valid index is treated as if it contains a value of a virtual type @defid{LUA\_TNONE}, which behaves like a nil value.

}

@sect3{constchar|@title{Pointers to strings}}

Several functions in the API return pointers (@T{const char\*}) to Lua strings in the stack. (See @Lid{lua\_pushfstring}, @Lid{lua\_pushlstring}, @Lid{lua\_pushstring}, and @Lid{lua\_tolstring}. See also @Lid{luaL\_checklstring}, @Lid{luaL\_checkstring}, and @Lid{luaL\_tolstring} in the auxiliary library.)

In general,  
 Lua's garbage collection can free or move internal memory  
 and then invalidate pointers to  
 internal strings.  
 To allow a safe use of these pointers,  
 the API guarantees that any pointer to a string in a stack index  
 is valid while the string value at that index is not removed from the stack.  
 (It can be moved to another index, though.)  
 When the index is a pseudo-index (referring to an upvalue),  
 the pointer is valid while the corresponding call is active and  
 the corresponding upvalue is not modified.

Some functions in the debug interface  
 also return pointers to strings,  
 namely `@Lid{lua_getlocal}`, `@Lid{lua_getupvalue}`,  
`@Lid{lua_setlocal}`, and `@Lid{lua_setupvalue}`.  
 For these functions, the pointer is guaranteed to  
 be valid while the caller function is active and  
 the given closure (if one was given) is in the stack.

Except for these guarantees,  
 the garbage collector is free to invalidate  
 any pointer to internal strings.

}

}

`@sect2{c-closure| @title{C Closures}`

When a `@N{C function}` is created,  
 it is possible to associate some values with it,  
 thus creating a `@def{@N{C closure}}`  
`@seeC{lua_pushcclosure};`  
 these  
 values are called `@def{upvalues}` and are  
 accessible to the function whenever it is called.

Whenever a `@N{C function}` is called,  
 its upvalues are located at specific pseudo-indices.  
 These pseudo-indices are produced by the macro  
`@Lid{lua_upvalueindex}`.  
 The first upvalue associated with a function is at index  
`@T{lua_upvalueindex(1)}`, and so on.  
 Any access to `@T{lua_upvalueindex(@rep{n})}`,  
 where `@rep{n}` is greater than the number of upvalues of the  
 current function

(but not greater than 256,  
which is one plus the maximum number of upvalues in a closure),  
produces an acceptable but invalid index.

A `@N{C closure}` can also change the values  
of its corresponding upvalues.

}

`@sect2{registry| @title{Registry}}`

Lua provides a `@def{registry}`,  
a predefined table that can be used by any `@N{C code}` to  
store whatever Lua values it needs to store.

The registry table is always accessible at pseudo-index  
`@defid{LUA_REGISTRYINDEX}`.

Any `@N{C library}` can store data into  
this table,

but it must take care to choose keys  
that are different from those used  
by other libraries, to avoid collisions.

Typically, you should use as key a string containing your library name,  
or a light userdata with the address of a `@N{C object}` in your code,  
or any Lua object created by your code.

As with variable names,  
string keys starting with an underscore followed by  
uppercase letters are reserved for Lua.

The integer keys in the registry are used  
by the reference mechanism `@seeC{luaL_ref}`  
and by some predefined values.

Therefore, integer keys in the registry  
must not be used for other purposes.

When you create a new Lua state,  
its registry comes with some predefined values.

These predefined values are indexed with integer keys  
defined as constants in `@id{lua.h}`.

The following constants are defined:

`@description{`

`@item{@defid{LUA_RIDX_MAINTHREAD}|` At this index the registry has  
the main thread of the state.

(The main thread is the one created together with the state.)

`}`

`@item{@defid{LUA_RIDX_GLOBALS}|`

At this index the registry has  
the `@x{global environment}`.

```
}  
}  
  
}
```

@sect2{C-error|@title{Error Handling in C}}

@simplesect{

Internally, Lua uses the C @id{longjmp} facility to handle errors.

(Lua will use exceptions if you compile it as C++;

search for @id{LUA\_THROW} in the source code for details.)

When Lua faces any error,

such as a @x{memory allocation error} or a type error,

it @emph{raises} an error;

that is, it does a long jump.

A @emphx{protected environment} uses @id{setjmp}

to set a recovery point;

any error jumps to the most recent active recovery point.

Inside a @N{C function} you can raise an error explicitly  
by calling @Lid{lua\_error}.

Most functions in the API can raise an error,

for instance due to a @x{memory allocation error}.

The documentation for each function indicates whether  
it can raise errors.

If an error happens outside any protected environment,  
Lua calls a @def{panic function} (see @Lid{lua\_atpanic})  
and then calls @T{abort},  
thus

exiting the host application.

Your panic function can avoid this exit by  
never returning

(e.g., doing a long jump to your own recovery point outside Lua).

The panic function,  
as its name implies,  
is a mechanism of last resort.

Programs should avoid it.

As a general rule,

when a @N{C function} is called by Lua with a Lua state,  
it can do whatever it wants on that Lua state,  
as it should be already protected.

However,

when C code operates on other Lua states

(e.g., a Lua-state argument to the function,

a Lua state stored in the registry, or  
the result of `@Lid{lua_newthread}`),  
it should use them only in API calls that cannot raise errors.

The panic function runs as if it were a `@x{message handler} @see{error}`;  
in particular, the error object is on the top of the stack.  
However, there is no guarantee about stack space.  
To push anything on the stack,  
the panic function must first check the available space `@see{stacksize}`.

}

`@sect3{statuscodes|@title{Status Codes}`

Several functions  
that report errors in the API use the following  
status codes to indicate different kinds of errors or other conditions:  
`@description{`

`@item{@defid{LUA_OK} (0)| no errors.}`

`@item{@defid{LUA_ERRRUN}| a runtime error.}`

`@item{@defid{LUA_ERRMEM}|  
@x{memory allocation error}.  
For such errors, Lua does not call the @x{message handler}.  
}`

`@item{@defid{LUA_ERRERR}| error while running the @x{message handler}.}`

`@item{@defid{LUA_ERRSYNTAX}| syntax error during precompilation.}`

`@item{@defid{LUA_YIELD}| the thread (coroutine) yields.}`

`@item{@defid{LUA_ERRFILE}| a file-related error;  
e.g., it cannot open or read the file.}`

}

These constants are defined in the header file `@id{lua.h}`.

}

}

`@sect2{continuations|@title{Handling Yields in C}`

Internally, Lua uses the C `@id{longjmp}` facility to yield a coroutine.

Therefore, if a `@N{C function}` `@id{foo}` calls an API function and this API function yields (directly or indirectly by calling another function that yields), Lua cannot return to `@id{foo}` any more, because the `@id{longjmp}` removes its frame from the `@N{C stack}`.

To avoid this kind of problem, Lua raises an error whenever it tries to yield across an API call, except for three functions: `@Lid{lua_yieldk}`, `@Lid{lua_callk}`, and `@Lid{lua_pcallk}`. All those functions receive a `@def{continuation function}` (as a parameter named `@id{k}`) to continue execution after a yield.

We need to set some terminology to explain continuations. We have a `@N{C function}` called from Lua which we will call the `@emph{original function}`. This original function then calls one of those three functions in the C API, which we will call the `@emph{callee function}`, that then yields the current thread. This can happen when the callee function is `@Lid{lua_yieldk}`, or when the callee function is either `@Lid{lua_callk}` or `@Lid{lua_pcallk}` and the function called by them yields.

Suppose the running thread yields while executing the callee function. After the thread resumes, it eventually will finish running the callee function. However, the callee function cannot return to the original function, because its frame in the `@N{C stack}` was destroyed by the yield. Instead, Lua calls a `@def{continuation function}`, which was given as an argument to the callee function. As the name implies, the continuation function should continue the task of the original function.

As an illustration, consider the following function:

```
@verbatim{
int original_function (lua_State *L) {
...   /* code 1 */
status = lua_pcall(L, n, m, h); /* calls Lua */
...   /* code 2 */
}
}
```

Now we want to allow the Lua code being run by `@Lid{lua_pcall}` to yield. First, we can rewrite our function like here:



```
@verbatim{
int k (lua_State *L, int status, lua_KContext ctx) {
... /* code 2 */
}
```

```
int original_function (lua_State *L) {
... /* code 1 */
return k(L, lua_pcall(L, n, m, h), ctx);
}
}
```

In the above code,

the new function `@id{k}` is a

`@emph{continuation function}` (with type `@Lid{lua_KFunction}`),  
which

should do all the work that the original function  
was doing after calling `@Lid{lua_pcall}`.

Now, we must inform Lua that it must call `@id{k}` if the Lua code  
being executed by `@Lid{lua_pcall}` gets interrupted in some way  
(errors or yielding),

so we rewrite the code as here,

replacing `@Lid{lua_pcall}` by `@Lid{lua_pcallk}`:

```
@verbatim{
int original_function (lua_State *L) {
... /* code 1 */
return k(L, lua_pcallk(L, n, m, h, ctx2, k), ctx1);
}
}
```

Note the external, explicit call to the continuation:

Lua will call the continuation only if needed, that is,  
in case of errors or resuming after a yield.

If the called function returns normally without ever yielding,

`@Lid{lua_pcallk}` (and `@Lid{lua_callk}`) will also return normally.

(Of course, instead of calling the continuation in that case,  
you can do the equivalent work directly inside the original function.)

Besides the Lua state,

the continuation function has two other parameters:

the final status of the call and the context value

(`@id{ctx}`) that

was passed originally to `@Lid{lua_pcallk}`.

Lua does not use this context value;

it only passes this value from the original function to the  
continuation function.

For `@Lid{lua_pcallk}`,

the status is the same value that would be returned by `@Lid{lua_pcallk}`,  
except that it is `@Lid{LUA_YIELD}` when being executed after a yield  
(instead of `@Lid{LUA_OK}`).

For `@Lid{lua_yieldk}` and `@Lid{lua_callk}`,

the status is always `@Lid{LUA_YIELD}` when Lua calls the continuation.  
 (For these two functions,  
 Lua will not call the continuation in case of errors,  
 because they do not handle errors.)  
 Similarly, when using `@Lid{lua_callk}`,  
 you should call the continuation function  
 with `@Lid{LUA_OK}` as the status.  
 (For `@Lid{lua_yieldk}`, there is not much point in calling  
 directly the continuation function,  
 because `@Lid{lua_yieldk}` usually does not return.)

Lua treats the continuation function as if it were the original function.  
 The continuation function receives the same Lua stack  
 from the original  
 function,  
 in the same state it would be if the callee function had returned.  
 (For instance,  
 after a `@Lid{lua_callk}` the function and its arguments are  
 removed from the stack and replaced by the results from the call.)  
 It also has the same upvalues.  
 Whatever it returns is handled by Lua as if it were the return  
 of the original function.

}

@sect2{@title{Functions and Types}

Here we list all functions and types from the `@N{C API}` in  
 alphabetical order.

Each function has an indicator like this:

`@apii{o,p,x}`

The first field, `@T{o}`,  
 is how many elements the function pops from the stack.  
 The second field, `@T{p}`,  
 is how many elements the function pushes onto the stack.  
 (Any function always pushes its results after popping its arguments.)  
 A field in the form `@T{x|y}` means the function can push (or pop)  
`@T{x}` or `@T{y}` elements,  
 depending on the situation;  
 an interrogation mark `@Char{?}` means that  
 we cannot know how many elements the function pops/pushes  
 by looking only at its arguments.  
 (For  
 instance, they may depend on what is in the stack.)  
 The third field, `@T{x}`,  
 tells whether the function may raise errors:  
`@Char{-}` means the function never raises any error;

@Char{m} means the function may raise only out-of-memory errors;  
@Char{v} means the function may raise the errors explained in the text;  
@Char{e} means the function can run arbitrary Lua code,  
either directly or through metamethods,  
and therefore may raise any errors.

```
@APIEntry{int lua_absindex (lua_State *L, int idx);|  
@apii{0,0,-}}
```

Converts the @x{acceptable index} @id{idx}  
into an equivalent @x{absolute index}  
(that is, one that does not depend on the stack size).

```
}
```

```
@APIEntry{  
typedef void * (*lua_Alloc) (void *ud,  
                             void *ptr,  
                             size_t osize,  
                             size_t nsize);|
```

The type of the @x{memory-allocation function} used by Lua states.  
The allocator function must provide a  
functionality similar to @id{realloc},  
but not exactly the  
same.

Its arguments are

@id{ud}, an opaque pointer passed to @Lid{lua\_newstate};  
@id{ptr}, a pointer to the block being allocated/reallocated/freed;  
@id{osize}, the original size of the block or some code about what  
is being allocated;  
and @id{nsize}, the new size of the block.

When @id{ptr} is not @id{NULL},  
@id{osize} is the size of the block pointed by @id{ptr},  
that is, the size given when it was allocated or reallocated.

When @id{ptr} is @id{NULL},  
@id{osize} encodes the kind of object that Lua is allocating.  
@id{osize} is any of  
@Lid{LUA\_TSTRING}, @Lid{LUA\_TTABLE}, @Lid{LUA\_TFUNCTION},  
@Lid{LUA\_TUSERDATA}, or @Lid{LUA\_TTHREAD} when (and only when)  
Lua is creating a new object of that type.  
When @id{osize} is some other value,  
Lua is allocating memory for something else.

Lua assumes the following behavior from the allocator function:

When @id{nsiz} is zero,  
the allocator must behave like @id{free}  
and then return @id{NULL}.

When @id{nsiz} is not zero,  
the allocator must behave  
like @id{realloc}.  
In particular, the allocator returns @id{NULL}  
if and only if it cannot fulfill the request.

Here is a simple implementation for the @x{allocator function}.  
It is used in the auxiliary library by @Lid{luaL\_newstate}.

```
@verbatim{
static void *l_alloc (void *ud, void *ptr, size_t osize,
                      size_t nsiz) {
  (void)ud; (void)osize; /* not used */
  if (nsiz == 0) {
    free(ptr);
    return NULL;
  }
  else
    return realloc(ptr, nsiz);
}
}
```

Note that @N{ISO C} ensures  
that @T{free(NULL)} has no effect and that  
@T{realloc(NULL,size)} is equivalent to @T{malloc(size)}.

```
}
```

```
@APIEntry{void lua_arith (lua_State *L, int op);|
@apii{2|1,1,e}
```

Performs an arithmetic or bitwise operation over the two values  
(or one, in the case of negations)  
at the top of the stack,  
with the value on the top being the second operand,  
pops these values, and pushes the result of the operation.  
The function follows the semantics of the corresponding Lua  
operator  
(that is, it may call metamethods).

The value of @id{op} must be one of the following constants:  
@description{

@item{@defid{LUA\_OPADD}| performs addition (@T{+})}

```

@item{@defid{LUA_OPSUB}| performs subtraction (@T{-})}
@item{@defid{LUA_OPMUL}| performs multiplication (@T{*})}
@item{@defid{LUA_OPDIV}| performs float division (@T{/})}
@item{@defid{LUA_OPIDIV}| performs floor division (@T{//})}
@item{@defid{LUA_OPMOD}| performs modulo (@T{%})}
@item{@defid{LUA_OPPOW}| performs exponentiation (@T{^})}
@item{@defid{LUA_OPUNM}| performs mathematical negation (unary @T{-})}
@item{@defid{LUA_OPBNOT}| performs bitwise NOT (@T{~})}
@item{@defid{LUA_OPBAND}| performs bitwise AND (@T{&})}
@item{@defid{LUA_OPBOR}| performs bitwise OR (@T{|})}
@item{@defid{LUA_OPBXOR}| performs bitwise exclusive OR (@T{~})}
@item{@defid{LUA_OPSHL}| performs left shift (@T{<<})}
@item{@defid{LUA_OPSHR}| performs right shift (@T{>>})}

```

```

}

```

```

}

```

```

@APIEntry{lua_CFunction lua_atpanic (lua_State *L, lua_CFunction
panicf);|
@apii{0,0,-}

```

Sets a new panic function and returns the old one @see{C-error}.

```

}

```

```

@APIEntry{void lua_call (lua_State *L, int nargs, int nresults);|
@apii{nargs+1,nresults,e}

```

Calls a function.

Like regular Lua calls,

@id{lua\_call} respects the @idx{\_\_call} metamethod.

So, here the word @Q{function}

means any callable value.

To do a call you must use the following protocol:

first, the function to be called is pushed onto the stack;

then, the arguments to the call are pushed

in direct order;

that is, the first argument is pushed first.

Finally you call @Lid{lua\_call};

@id{nargs} is the number of arguments that you pushed onto the stack.

When the function returns,

all arguments and the function value are popped

and the call results are pushed onto the stack.

The number of results is adjusted to @id{nresults},

unless @id{nresults} is @defid{LUA\_MULTRET}.

In this case, all results from the function are pushed;

Lua takes care that the returned values fit into the stack space,  
but  
it does not ensure any extra space in the stack.  
The function results are pushed onto the stack in direct order  
(the first result is pushed first),  
so that after the call the last result is on the top of the stack.

Any error while calling and running the function is propagated upwards  
(with a `@id{longjmp}`).

The following example shows how the host program can do the  
equivalent to this Lua code:

```
@verbatim{
```

```
a = f("how", t.x, 14)
}
```

Here it is `@N{in C}`:

```
@verbatim{
```

```
lua_getglobal(L, "f");          /* function to be called */
lua_pushliteral(L, "how");      /* 1st argument */
lua_getglobal(L, "t");          /* table to be indexed */
lua_getfield(L, -1, "x");        /* push result of t.x (2nd arg) */
lua_remove(L, -2);              /* remove 't' from the stack */
lua_pushinteger(L, 14);         /* 3rd argument */
lua_call(L, 3, 1);              /* call 'f' with 3 arguments and 1 result */
lua_setglobal(L, "a");          /* set global
'a' */
}
```

Note that the code above is `@emph{balanced}`:

at its end, the stack is back to its original configuration.

This is considered good programming practice.

```
}
```

```
@APIEntry{
```

```
void lua_callk (lua_State *L,
               int nargs,
               int nresults,
               lua_KContext ctx,
               lua_KFunction k);|
@apii{nargs + 1,nresults,e}
```

This function behaves exactly like `@Lid{lua_call}`,

but allows the called function to yield `@see{continuations}`.

```
}
```

```
@APIEntry{typedef int (*lua_CFunction) (lua_State *L);|
```

Type for @N{C functions}.

In order to communicate properly with Lua,  
a @N{C function} must use the following protocol,  
which defines the way parameters and results are passed:  
a @N{C function} receives its arguments from Lua in its stack  
in direct order (the first argument is pushed first).  
So, when the function starts,  
@T{lua\_gettop(L)} returns the number of arguments received by the function.  
The first argument (if any) is at index 1  
and its last argument  
is at index @T{lua\_gettop(L)}.  
To return values to Lua, a @N{C function} just pushes them onto the stack,  
in direct order (the first result is pushed first),  
and returns in C the number of results.  
Any other value in the stack below the results will be properly  
discarded by Lua.  
Like a Lua function, a @N{C function} called by Lua can also return  
many results.

As an example, the following function receives a variable number  
of numeric arguments and returns their average and their sum:

```
@verbatim{
static int foo (lua_State *L) {
    int n = lua_gettop(L); /* number of arguments */
    lua_Number sum = 0.0;
    int i;
    for (i = 1; i <= n; i++) {
        if (!lua_isnumber(L, i)) {
            lua_pushliteral(L, "incorrect argument");
            lua_error(L);
        }
        sum += lua_tonumber(L, i);
    }
    lua_pushnumber(L, sum/n); /* first result */
    lua_pushnumber(L, sum); /* second result */
    return 2; /* number of results */
}
}
```

```
@APIEntry{int lua_checkstack (lua_State
*L, int n);|
@api{0,0,-}
```

Ensures that the stack has space for at least @id{n} extra elements, that is, that you can safely push up to @id{n} values into it. It returns false if it cannot fulfill the request, either because it would cause the stack to be greater than a fixed maximum size (typically at least several thousand elements) or because it cannot allocate memory for the extra space. This function never shrinks the stack; if the stack already has space for the extra elements, it is left unchanged.

}

```
@APIEntry{void lua_close (lua_State *L);|
@apii{0,0,-}}
```

Close all active to-be-closed variables in the main thread, release all objects in the given Lua state (calling the corresponding garbage-collection metamethods, if any), and frees all dynamic memory used by this state.

On several platforms, you may not need to call this function, because all resources are naturally released when the host program ends. On the other hand, long-running programs that create multiple states, such as daemons or web servers, will probably need to close states as soon as they are not needed.

}

```
@APIEntry{void lua_closeslot (lua_State *L, int index);|
@apii{0,0,e}}
```

Close the to-be-closed slot at the given index and set its value to @nil. The index must be the last index previously marked to be closed @see{lua\_toclose} that is still active (that is, not closed yet).

A @idx{\_\_close} metamethod cannot yield when called through this function.

(This function was introduced in @N{release 5.4.3}.)

}

```
@APIEntry{int lua_closethread (lua_State *L, lua_State *from);|
@apii{0,?, -}}
```



Resets a thread, cleaning its call stack and closing all pending to-be-closed variables.

Returns a status code:

@Lid{LUA\_OK} for no errors in the thread  
(either the original error that stopped the thread or errors in closing methods),  
or an error status otherwise.

In case of error,

leaves the error object on the top of the stack.

The parameter @id{from} represents the coroutine that is resetting @id{L}.

If

there is no such coroutine,  
this parameter can be @id{NULL}.

(This function was introduced in @N{release 5.4.6}.)

}

@APIEntry{int lua\_compare (lua\_State \*L, int index1, int index2, int op);|  
@apii{0,0,e}}

Compares two Lua values.

Returns 1 if the value at index @id{index1} satisfies @id{op}  
when compared with the value at index @id{index2},  
following the semantics of the corresponding Lua operator  
(that is, it may call metamethods).

Otherwise @N{returns 0}.

Also @N{returns 0} if any of the indices is not valid.

The value of @id{op} must be one of the following constants:

@description{

@item{@defid{LUA\_OPEQ}| compares for equality (@T{==})}

@item{@defid{LUA\_OPLT}| compares for less than (@T{<})}

@item{@defid{LUA\_OPLE}| compares for less or equal (@T{<=})}

}

}

@APIEntry{void lua\_concat (lua\_State \*L, int n);|  
@apii{n,1,e}}

Concatenates the @id{n} values at the top of the stack,  
pops them, and leaves the result on the top.

If @N{@T{n} is 1}, the result is the single value on the

stack  
(that is, the function does nothing);  
if @id{n} is 0, the result is the empty string.  
Concatenation is performed following the usual semantics of Lua  
@see{concat}.

}

@APIEntry{void lua\_copy (lua\_State \*L, int fromidx, int toidx);|  
@apii{0,0,-}}

Copies the element at index @id{fromidx}  
into the valid index @id{toidx},  
replacing the value at that position.  
Values at other positions are not affected.

}

@APIEntry{void lua\_createtable (lua\_State \*L, int narr, int nrec);|  
@apii{0,1,m}}

Creates a new empty table and pushes it onto the stack.  
Parameter @id{narr} is a hint for how many elements the table  
will have as a sequence;  
parameter @id{nrec} is a hint for how many other elements  
the table will have.  
Lua may use these hints to preallocate memory for the new table.  
This preallocation may help performance when you know in advance  
how many elements the table will have.  
Otherwise you can use the function @Lid{lua\_newtable}.

}

@APIEntry{int lua\_dump (lua\_State \*L,

lua\_Writer writer,  
void \*data,  
int strip);|

@apii{0,0,-}}

Dumps a function as a binary chunk.  
Receives a Lua function on the top of the stack  
and produces a binary chunk that,  
if loaded again,  
results in a function equivalent to the one dumped.  
As it produces parts of the chunk,  
@Lid{lua\_dump} calls function @id{writer} @seeC{lua\_Writer}  
with the given @id{data}

to write them.

If `@id{strip}` is true,  
the binary representation may not include all debug information  
about the function,  
to save space.

The value returned is the error code returned by the last  
call to the writer;  
`@N{0 means}` no errors.

This function does not pop the Lua function from the stack.

}

`@APIEntry{int lua_error (lua_State *L);|`  
`@apii{1,0,v}`

Raises a Lua error,  
using the value on the top of the stack as the error object.  
This function does a long jump,  
and therefore never returns  
`@seeC{luaL_error}`.

}

`@APIEntry{int lua_gc (lua_State *L, int what, ...);|`  
`@apii{0,0,-}`

Controls  
the garbage collector.

This function performs several tasks,  
according to the value of the parameter `@id{what}`.  
For options that need extra arguments,  
they are listed after the option.  
`@description{`

`@item{@id{LUA_GCCOLLECT}}|`  
Performs a full garbage-collection cycle.  
}

`@item{@id{LUA_GCSTOP}}|`  
Stops the garbage collector.  
}

`@item{@id{LUA_GCRESTART}}|`  
Restarts the garbage collector.

}

@item{@id{LUA\_GCCOUNT}|

Returns the current amount of memory (in Kbytes) in use by Lua.

}

@item{@id{LUA\_GCCOUNTB}|

Returns the remainder of dividing the current amount of bytes of memory in use by Lua by 1024.

}

@item{@id{LUA\_GCSTEP} @T{(int stepsize)}|

Performs an incremental step of garbage collection, corresponding to the allocation of @id{stepsize} Kbytes.

}

@item{@id{LUA\_GCISRUNNING}|

Returns a boolean that tells whether the collector is running (i.e., not stopped).

}

@item{@id{LUA\_GCINC} (int pause, int stepmul, stepsize)|

Changes the collector to incremental mode

with the

given parameters @see{incmode}.

Returns the previous mode (@id{LUA\_GCGEN} or @id{LUA\_GCINC}).

}

@item{@id{LUA\_GCGEN} (int minormul, int majormul)|

Changes the collector to generational mode

with the given parameters @see{genmode}.

Returns the previous mode (@id{LUA\_GCGEN} or @id{LUA\_GCINC}).

}

}

For more details about these options,

see @Lid{collectgarbage}.

This function should not be called by a finalizer.

}

@APIEntry{lua\_Alloc lua\_getallocf (lua\_State \*L, void \*\*ud);|

@apii{0,0,-}

Returns the @x{memory-allocation function} of a given state.

If @id{ud} is not @id{NULL}, Lua stores in @T{\*ud} the

opaque pointer given when the memory-allocator function was set.

```
}
```

```
@APIEntry{int lua_getfield (lua_State *L, int index, const char *k);|
@apii{0,1,e}
```

Pushes onto the stack the value `@T{t[k]}`,  
where `@id{t}` is the value at the given index.  
As in Lua, this function may trigger a metamethod  
for the `@Q{index}` event `@see{metatable}`.

Returns the type of the pushed value.

```
}
```

```
@APIEntry{void
*lua_getextraspace (lua_State *L);|
@apii{0,0,-}
```

Returns a pointer to a raw memory area associated with the  
given Lua state.  
The application can use this area for any purpose;  
Lua does not use it for anything.

Each new thread has this area initialized with a copy  
of the area of the `@x{main thread}`.

By default, this area has the size of a pointer to void,  
but you can recompile Lua with a different size for this area.  
(See `@id{LUA_EXTRASPACE}` in `@id{luaconf.h}`.)

```
}
```

```
@APIEntry{int lua_getglobal (lua_State *L, const char *name);|
@apii{0,1,e}
```

Pushes onto the stack the value of the global `@id{name}`.  
Returns the type of that value.

```
}
```

```
@APIEntry{int lua_geti (lua_State *L, int index, lua_Integer i);|
@apii{0,1,e}
```

Pushes onto the stack the value `@T{t[i]}`,  
where `@id{t}` is the value at the given index.  
As in Lua, this function may trigger a metamethod  
for the `@Q{index}` event `@see{metatable}`.

Returns the type of the pushed value.

}

```
@APIEntry{int lua_getmetatable (lua_State *L, int index);|
@apii{0,0|1,-}}
```

If

the value at the given index has a metatable,  
the function pushes that metatable onto the stack and @N{returns 1}.

Otherwise,

the function @N{returns 0} and pushes nothing on the stack.

}

```
@APIEntry{int lua_gettable (lua_State *L, int index);|
@apii{1,1,e}}
```

Pushes onto the stack the value @T{t[k]},  
where @id{t} is the value at the given index  
and @id{k} is the value on the top of the stack.

This function pops the key from the stack,  
pushing the resulting value in its place.  
As in Lua, this function may trigger a metamethod  
for the @Q{index} event @see{metatable}.

Returns the type of the pushed value.

}

```
@APIEntry{int lua_gettop (lua_State *L);|
@apii{0,0,-}}
```

Returns the index of the top element in the stack.  
Because indices start @N{at 1},  
this result is equal to the number of elements in the stack;  
in particular, @N{0 means} an empty stack.

}

```
@APIEntry{int lua_getiuservalue (lua_State *L, int index, int n);|
@apii{0,1,-}}
```

Pushes onto the stack the @id{n}-th  
user value associated with the  
full userdata at the given index and

returns the type of the pushed value.

If the userdata does not have that value,  
pushes @nil and returns @Lid{LUA\_TNONE}.

}

@APIEntry{void lua\_insert (lua\_State \*L, int index);|  
@apii{1,1,-}}

Moves the top element into the given valid index,  
shifting up the elements above this index to open space.  
This function cannot be called with a pseudo-index,  
because a pseudo-index is not an actual stack position.

}

@APIEntry{typedef @ldots lua\_Integer;|

The type of integers in Lua.

By default this type is @id{long long},  
(usually a 64-bit two-complement integer),  
but that can be changed to @id{long} or @id{int}  
(usually a 32-bit two-complement integer).  
(See @id{LUA\_INT\_TYPE} in @id{luaconf.h}.)

Lua also defines the constants

@defid{LUA\_MININTEGER} and @defid{LUA\_MAXINTEGER},  
with the minimum and the maximum values that fit in this type.

}

@APIEntry{int lua\_isboolean (lua\_State \*L, int index);|  
@apii{0,0,-}}

Returns 1  
if the value at the given index is a boolean,  
and @N{0 otherwise}.

}

@APIEntry{int lua\_isfunction (lua\_State \*L, int index);|  
@apii{0,0,-}}

Returns 1 if the value at the given index is a @N{C function},  
and @N{0 otherwise}.

}

```
@APIEntry{int lua_isfunction (lua_State *L, int index);|
@apii{0,0,-}}
```

Returns 1 if the value at the given index is a function  
(either C or Lua), and @N{0 otherwise}.

}

```
@APIEntry{int lua_isinteger (lua_State *L, int index);|
@apii{0,0,-}}
```

Returns 1 if the value at the given index is an integer  
(that is, the value is a number and is represented as an integer),  
and @N{0 otherwise}.

}

```
@APIEntry{int lua_islightuserdata (lua_State *L, int index);|
@apii{0,0,-}}
```

Returns 1 if the value at the given index is a light userdata,  
and @N{0 otherwise}.

}

```
@APIEntry{int lua_isnil (lua_State *L, int index);|
@apii{0,0,-}}
```

Returns 1 if the value at the given index is @nil,  
and @N{0 otherwise}.

}

```
@APIEntry{int lua_isnone (lua_State *L, int index);|
@apii{0,0,-}}
```

Returns  
1 if the given index is not valid,  
and @N{0 otherwise}.

}

```
@APIEntry{int lua_isnoneornil (lua_State *L, int index);|
@apii{0,0,-}}
```

Returns 1 if the given index is not valid



or if the value at this index is @nil,  
and @N{0 otherwise}.

}

@APIEntry{int lua\_isnumber (lua\_State \*L, int index);|  
@apii{0,0,-}}

Returns 1 if the value at the given index is a number  
or a string convertible to a number,  
and @N{0 otherwise}.

}

@APIEntry{int lua\_isstring (lua\_State \*L, int index);|  
@apii{0,0,-}}

Returns 1 if the value at the given index is a string  
or a number (which is always convertible to a string),  
and @N{0 otherwise}.

}

@APIEntry{int lua\_istable (lua\_State \*L, int index);|  
@apii{0,0,-}}

Returns 1 if the value at the given index is a table,  
and @N{0 otherwise}.

}

@APIEntry{int lua\_isthread (lua\_State \*L, int index);|  
@apii{0,0,-}}

Returns 1 if the value at the given index is a thread,  
and @N{0 otherwise}.

}

@APIEntry{int lua\_isuserdata (lua\_State \*L, int index);|  
@apii{0,0,-}}

Returns  
1 if the value at the given index is a userdata  
(either full or light), and @N{0 otherwise}.

}

```
@APIEntry{int lua_isyieldable (lua_State *L);|
@apii{0,0,-}}
```

Returns 1 if the given coroutine can yield,  
and @N{0 otherwise}.

```
}
```

```
@APIEntry{typedef @ldots lua_KContext;|
```

The type for continuation-function contexts.

It must be a numeric type.

This type is defined as @id{intptr\_t}

when @id{intptr\_t} is available,

so that it can store pointers too.

Otherwise, it is defined as @id{ptrdiff\_t}.

```
}
```

```
@APIEntry{
typedef int (*lua_KFunction) (lua_State *L, int status, lua_KContext ctx);|
```

Type for continuation functions @see{continuations}.

```
}
```

```
@APIEntry{void lua_len (lua_State *L, int index);|
@apii{0,1,e}}
```

Returns the length of the value at the given index.

It is equivalent to the @Char{#} operator in Lua @see{len-op} and  
may trigger a metamethod for the @Q{length} event @see{metatable}.

The result is pushed on the stack.

```
}
```

```
@APIEntry{
int lua_load (lua_State *L,
              lua_Reader reader,
              void *data,
              const char *chunkname,
              const char *mode);|
@apii{0,1,-}}
```

Loads a Lua chunk without running it.

If there are no errors,

@id{lua\_load} pushes the compiled chunk as a Lua  
function on top of the stack.

Otherwise, it pushes an error message.

The `@id{lua_load}` function uses a user-supplied `@id{reader}` function to read the chunk `@seeC{lua_Reader}`.

The `@id{data}` argument is an opaque value passed to the reader function.

The `@id{chunkname}` argument gives a name to the chunk, which is used for error messages and in debug information `@see{debugI}`.

`@id{lua_load}` automatically detects whether the chunk is text or binary and loads it accordingly (see program `@idx{luac}`).

The string `@id{mode}` works as in function `@Lid{load}`, with the addition that

a `@id{NULL}` value is equivalent to the string `@St{bt}`.

`@id{lua_load}` uses the stack internally, so the reader function must always leave the stack unmodified when returning.

`@id{lua_load}` can return

`@Lid{LUA_OK}`,

`@Lid{LUA_ERRSYNTAX}`, or `@Lid{LUA_ERRMEM}`.

The function may also return other values corresponding to errors raised by the read function `@see{statuscodes}`.

If the resulting function has upvalues, its first upvalue is set to the value of the `@x{global environment}` stored at index `@id{LUA_RIDX_GLOBALS}` in the registry `@see{registry}`.

When loading main chunks,

this upvalue will be the `@id{_ENV}` variable `@see{globalenv}`.

Other upvalues are initialized with `@nil`.

}

`@APIEntry{lua_State *lua_newstate (lua_Alloc f, void *ud);`

`@apii{0,0,-}`

Creates a new independent state and returns its main thread.

Returns `@id{NULL}` if it cannot create the state (due to lack of memory).

The argument `@id{f}` is the `@x{allocator function}`;

Lua will do all memory allocation for this state through this function `@seeF{lua_Alloc}`.

The second argument, `@id{ud}`, is an opaque pointer that Lua passes to the allocator in every call.

}

```
@APIEntry{void lua_newtable (lua_State *L);|
@apii{0,1,m}}
```

Creates

a new empty table and pushes it onto the stack.

It is equivalent to @T{lua\_createtable(L, 0, 0)}.

```
}
```

```
@APIEntry{lua_State *lua_newthread (lua_State *L);|
@apii{0,1,m}}
```

Creates a new thread, pushes it on the stack,

and returns a pointer to a @Lid{lua\_State} that represents this new thread.

The new thread returned by this function shares with the original thread

its global environment,

but has an independent execution stack.

Threads are subject to garbage collection,

like any Lua object.

```
}
```

```
@APIEntry{void *lua_newuserdatauv (lua_State *L, size_t size, int nuvalue);|
@apii{0,1,m}}
```

This function creates and pushes on the stack a new full userdata,

with @id{nuvalue} associated Lua values, called @id{user values},

plus an associated block of raw memory with @id{size} bytes.

(The user values can be set and read with the functions

@Lid{lua\_setiuservalue} and @Lid{lua\_getiuservalue}.)

The function returns the address of the block of memory.

Lua ensures that this address is valid as long as

the

corresponding userdata is alive @see{GC}.

Moreover, if the userdata is marked for finalization @see{finalizers},

its address is valid at least until the call to its finalizer.

```
}
```

```
@APIEntry{int lua_next (lua_State *L, int index);|
@apii{1,2|0,v}}
```

Pops a key from the stack,

and pushes a key@En{ }value pair from the table at the given index,

the @Q{next} pair after the given key.

If there are no more elements in the table,

then @Lid{lua\_next} @N{returns 0} and pushes nothing.

A typical table traversal looks like this:

```
@verbatim{
/* table is in the stack at index 't' */
lua_pushnil(L); /* first key */
while (lua_next(L, t) != 0) {
/* uses 'key' (at index -2) and 'value' (at index -1) */
printf("%s - %s\n",
    lua_typename(L, lua_type(L, -2)),
    lua_typename(L, lua_type(L, -1)));
/* removes 'value'; keeps 'key' for next iteration */
lua_pop(L, 1);
}
}
```

While traversing a table,  
avoid calling @Lid{lua\_tolstring} directly on a key,  
unless you know that the key is  
actually a string.  
Recall that @Lid{lua\_tolstring} may change  
the value at the given index;  
this confuses the next call to @Lid{lua\_next}.

This function may raise an error if the given key  
is neither @nil nor present in the table.  
See function @Lid{next} for the caveats of modifying  
the table during its traversal.

```
}
```

```
@APIEntry{typedef @ldots lua_Number;|
```

The type of floats in Lua.

By default this type is double,  
but that can be changed to a single float or a long double.  
(See @id{LUA\_FLOAT\_TYPE} in @id{luaconf.h}.)

```
}
```

```
@APIEntry{int lua_numbertointeger (lua_Number n, lua_Integer *p);|
```

Tries to convert a Lua float to a Lua integer;  
the float @id{n} must have an integral value.  
If that value is within the range of Lua integers,  
it is converted to an integer and assigned to @T{\*p}.  
The macro results in a boolean indicating whether the

conversion was successful.

(Note that this range test can be tricky to do correctly without this macro, due to rounding.)

This macro may evaluate its arguments more than once.

}

```
@APIEntry{int lua_pcall (lua_State *L, int nargs, int nresults, int msgh);|
@apii{nargs + 1,nresults|1,-}
```

Calls a function (or a callable object) in protected mode.

Both @id{nargs} and @id{nresults} have the same meaning as in @Lid{lua\_call}.

If there are no errors during the call,

@Lid{lua\_pcall} behaves exactly like @Lid{lua\_call}.

However, if there is any error,

@Lid{lua\_pcall} catches it,

pushes a single value on the stack (the error object),

and returns an error code.

Like @Lid{lua\_call},

@Lid{lua\_pcall} always removes the function

and its arguments from the stack.

If @id{msgh} is 0,

then the error object returned on the stack

is exactly the original error object.

Otherwise, @id{msgh} is the stack index of a

@emph{message handler}.

(This index cannot be a pseudo-index.)

In case of runtime errors,

this handler will be called with the error object

and its return value will be the object

returned on the stack by @Lid{lua\_pcall}.

Typically, the message

handler is used to add more debug

information to the error object, such as a stack traceback.

Such information cannot be gathered after the return of @Lid{lua\_pcall},

since by then the stack has unwound.

The @Lid{lua\_pcall} function returns one of the following status codes:

@Lid{LUA\_OK}, @Lid{LUA\_ERRRUN}, @Lid{LUA\_ERRMEM}, or @Lid{LUA\_ERRERR}.

}

```
@APIEntry{
int lua_pcallk (lua_State *L,
               int nargs,
               int nresults,
               int msgch,
               lua_KContext ctx,
               lua_KFunction k);|
@apii{nargs + 1,nresults|1,-}
```

This function behaves exactly like @Lid{lua\_pcall},  
except that it allows the called function to yield @see{continuations}.

```
}
```

```
@APIEntry{void lua_pop (lua_State *L, int n);|
@apii{n,0,e}
```

Pops @id{n} elements from the stack.  
It is implemented as a macro over @Lid{lua\_settop}.

```
}
```

```
@APIEntry{void lua_pushboolean (lua_State *L, int b);|
@apii{0,1,-}
```

Pushes a boolean value with value @id{b} onto the stack.

```
}
```

```
@APIEntry{void
lua_pushcclosure (lua_State *L, lua_CFunction fn, int n);|
@apii{n,1,m}
```

Pushes a new @N{C closure} onto the stack.  
This function receives a pointer to a @N{C function}  
and pushes onto the stack a Lua value of type @id{function} that,  
when called, invokes the corresponding @N{C function}.  
The parameter @id{n} tells how many upvalues this function will have  
@see{c-closure}.

Any function to be callable by Lua must  
follow the correct protocol to receive its parameters  
and return its results @seeC{lua\_CFunction}.

When a @N{C function} is created,  
it is possible to associate some values with it,  
the so called upvalues;  
these upvalues are then accessible to the function whenever it is called.

This association is called a `@x{@N{C closure}}` `@see{c-closure}`.  
 To create a `@N{C closure}`,  
 first the initial values for its upvalues must be pushed onto the stack.  
 (When there are multiple upvalues, the first value is pushed first.)  
 Then `@Lid{lua_pushcclosure}`  
 is called to create and push the `@N{C`  
`function}` onto the stack,  
 with the argument `@id{n}` telling how many values will be  
 associated with the function.  
`@Lid{lua_pushcclosure}` also pops these values from the stack.

The maximum value for `@id{n}` is 255.

When `@id{n}` is zero,  
 this function creates a `@def{light @N{C function}}`,  
 which is just a pointer to the `@N{C function}`.  
 In that case, it never raises a memory error.

}

```
@APIEntry{void lua_pushcfunction (lua_State *L, lua_CFunction f);|
@apii{0,1,-}}
```

Pushes a `@N{C function}` onto the stack.  
 This function is equivalent to `@Lid{lua_pushcclosure}` with no upvalues.

}

```
@APIEntry{const char *lua_pushfstring (lua_State *L, const char *fmt, ...);|
@apii{0,1,v}}
```

Pushes onto the stack a formatted string  
 and returns a pointer to this string `@see{constchar}`.  
 It is similar to the `@ANSI{sprintf}`,  
 but has two important differences.  
 First,  
 you do not have to allocate space for the result;  
 the result is a Lua string and Lua takes care of memory allocation  
 (and deallocation, through garbage  
 collection).  
 Second,  
 the conversion specifiers are quite restricted.  
 There are no flags, widths, or precisions.  
 The conversion specifiers can only be  
`@Char{%%}` (inserts the character `@Char{%}`),  
`@Char{%s}` (inserts a zero-terminated string, with no size restrictions),  
`@Char{%f}` (inserts a `@Lid{lua_Number}`),  
`@Char{%I}` (inserts a `@Lid{lua_Integer}`),



`@Char{%p}` (inserts a pointer),  
`@Char{%d}` (inserts an `@T{int}`),  
`@Char{%c}` (inserts an `@T{int}` as a one-byte character), and  
`@Char{%U}` (inserts a `@T{long int}` as a `@x{UTF-8}` byte sequence).

This function may raise errors due to memory overflow  
 or an invalid conversion specifier.

}

`@APIEntry{void lua_pushglobaltable (lua_State *L);|`  
`@apii{0,1,-}`

Pushes the `@x{global environment}` onto the stack.

}

`@APIEntry{void lua_pushinteger (lua_State *L, lua_Integer n);|`  
`@apii{0,1,-}`

Pushes an integer with value `@id{n}` onto the stack.

}

`@APIEntry{void lua_pushlightuserdata (lua_State *L, void *p);|`  
`@apii{0,1,-}`

Pushes a light userdata onto the stack.

Userdata

represent `@N{C values}` in Lua.

A `@def{light userdata}` represents a pointer, a `@T{void*}`.

It is a value (like a number):

you do not create it, it has no individual metatable,

and it is not collected (as it was never created).

A light userdata is equal to `@Q{any}`

light userdata with the same `@N{C address}`.

}

`@APIEntry{const char *lua_pushliteral (lua_State *L, const char *s);|`  
`@apii{0,1,m}`

This macro is equivalent to `@Lid{lua_pushstring}`,  
 but should be used only when `@id{s}` is a literal string.  
 (Lua may optimize this case.)

}

```
@APIEntry{const char *lua_pushlstring (lua_State *L, const char *s, size_t len);|
@apii{0,1,m}
```

Pushes the string pointed to by @id{s} with size @id{len} onto the stack.

Lua will make or reuse an internal copy of the given string, so the memory at @id{s} can be freed or reused immediately after the function returns.

The string can contain any binary data, including @x{embedded zeros}.

Returns a pointer to the internal copy of the string @see{constchar}.

```
}
```

```
@APIEntry{void
lua_pushnil (lua_State *L);|
@apii{0,1,-}
```

Pushes a nil value onto the stack.

```
}
```

```
@APIEntry{void lua_pushnumber (lua_State *L, lua_Number n);|
@apii{0,1,-}
```

Pushes a float with value @id{n} onto the stack.

```
}
```

```
@APIEntry{const char *lua_pushstring (lua_State *L, const char *s);|
@apii{0,1,m}
```

Pushes the zero-terminated string pointed to by @id{s} onto the stack.

Lua will make or reuse an internal copy of the given string, so the memory at @id{s} can be freed or reused immediately after the function returns.

Returns a pointer to the internal copy of the string @see{constchar}.

If @id{s} is @id{NULL}, pushes @nil and returns @id{NULL}.

```
}
```

```
@APIEntry{int lua_pushthread (lua_State *L);|
@apii{0,1,-}
```

Pushes the thread represented by @id{L} onto the stack.  
Returns 1 if this thread is the @x{main thread} of its state.

}

@APIEntry{void lua\_pushvalue (lua\_State \*L, int index);|  
@apii{0,1,-}}

Pushes a copy of the element at the given index  
onto the stack.

}

@APIEntry{  
const  
char \*lua\_pushvfstring (lua\_State \*L,  
                        const char \*fmt,  
                        va\_list argp);|  
@apii{0,1,v}}

Equivalent to @Lid{lua\_pushfstring}, except that it receives a @id{va\_list}  
instead of a variable number of arguments.

}

@APIEntry{int lua\_rawequal (lua\_State \*L, int index1, int index2);|  
@apii{0,0,-}}

Returns 1 if the two values in indices @id{index1} and  
@id{index2} are primitively equal  
(that is, equal without calling the @idx{\_\_eq} metamethod).  
Otherwise @N{returns 0}.  
Also @N{returns 0} if any of the indices are not valid.

}

@APIEntry{int lua\_rawget (lua\_State \*L, int index);|  
@apii{1,1,-}}

Similar to @Lid{lua\_gettable}, but does a raw access  
(i.e., without metamethods).  
The value at @id{index} must be a table.

}

@APIEntry{int lua\_rawgeti (lua\_State \*L, int index, lua\_Integer n);|  
@apii{0,1,-}}

Pushes onto the stack the value `@T{t[n]}`,  
where `@id{t}` is the table at the given index.  
The access is raw,  
that is, it does not use  
the `@idx{__index}` metavalue.

Returns the type of the pushed value.

}

```
@APIEntry{int lua_rawgetp (lua_State *L, int index, const void *p);|
@apii{0,1,-}}
```

Pushes onto the stack the value `@T{t[k]}`,  
where `@id{t}` is the table at the given index and  
`@id{k}` is the pointer `@id{p}` represented as a light userdata.  
The access is raw;  
that is, it does not use the `@idx{__index}` metavalue.

Returns the type of the pushed value.

}

```
@APIEntry{lua_Unsigned lua_rawlen (lua_State *L, int index);|
@apii{0,0,-}}
```

Returns the raw `@Q{length}` of the value at the given index:  
for strings, this is the string length;  
for tables, this is the result of the length operator (`@Char{#}`)  
with no metamethods;  
for userdata, this is the size of the block of memory allocated  
for the userdata.  
For other values, this call `@N{returns 0}`.

}

```
@APIEntry{void lua_rawset (lua_State *L, int index);|
@apii{2,0,m}}
```

Similar to `@Lid{lua_settable}`, but does a raw assignment  
(i.e., without metamethods).  
The value at `@id{index}` must  
be a table.

}

```
@APIEntry{void lua_rawseti (lua_State *L, int index, lua_Integer i);|
```

```
@apii{1,0,m}
```

Does the equivalent of `@T{t[i] = v}`,  
where `@id{t}` is the table at the given index  
and `@id{v}` is the value on the top of the stack.

This function pops the value from the stack.  
The assignment is raw,  
that is, it does not use the `@idx{__newindex}` metavalue.

```
}
```

```
@APIEntry{void lua_rawsetp (lua_State *L, int index, const void *p);|  
@apii{1,0,m}
```

Does the equivalent of `@T{t[p] = v}`,  
where `@id{t}` is the table at the given index,  
`@id{p}` is encoded as a light userdata,  
and `@id{v}` is the value on the top of the stack.

This function pops the value from the stack.  
The assignment is raw,  
that is, it does not use the `@idx{__newindex}` metavalue.

```
}
```

```
@APIEntry{  
typedef const char * (*lua_Reader) (lua_State *L,  
                                     void *data,  
                                     size_t *size);|
```

The reader function used by `@Lid{lua_load}`.  
Every time `@Lid{lua_load}` needs another piece of  
the chunk,  
it calls the reader,  
passing along its `@id{data}` parameter.  
The reader must return a pointer to a block of memory  
with a new piece of the chunk  
and set `@id{size}` to the block size.  
The block must exist until the reader function is called again.  
To signal the end of the chunk,  
the reader must return `@id{NULL}` or set `@id{size}` to zero.  
The reader function may return pieces of any size greater than zero.

```
}
```

```
@APIEntry{void lua_register (lua_State *L, const char *name, lua_CFunction f);|  
@apii{0,0,e}
```

Sets the @N{C function} @id{f} as the new value of global @id{name}.

It is defined as a macro:

```
@verbatim{
#define lua_register(L,n,f) \
    (lua_pushcfunction(L, f), lua_setglobal(L, n))
}

}
```

```
@APIEntry{void lua_remove (lua_State *L, int index);|
```

```
@apii{1,0,-}
```

Removes the element at the given valid index,  
shifting down the elements above this index to fill the gap.

This function cannot be called with a pseudo-index,  
because a pseudo-index is not an actual stack position.

```
}
```

```
@APIEntry{void
```

```
lua_replace (lua_State *L, int index);|
```

```
@apii{1,0,-}
```

Moves the top element into the given valid index  
without shifting any element  
(therefore replacing the value at that given index),  
and then pops the top element.

```
}
```

```
@APIEntry{int lua_resetthread (lua_State *L);|
```

```
@apii{0,?, -}
```

This function is deprecated;  
it is equivalent to @Lid{lua\_closethread} with  
@id{from} being @id{NULL}.

```
}
```

```
@APIEntry{int lua_resume (lua_State *L, lua_State *from, int nargs,
                        int *nresults);|
```

```
@apii{?,?, -}
```

Starts and resumes a coroutine in the given thread @id{L}.

To start a coroutine,  
you push the main function plus any arguments

onto the empty stack of the thread.  
 then you call `@Lid{lua_resume}`,  
 with `@id{nargs}` being the number of arguments.  
 This call returns when the coroutine suspends or finishes its execution.  
 When it returns,  
`@id{*nresults}` is updated and  
 the top of the stack contains  
 the `@id{*nresults}` values passed to `@Lid{lua_yield}`  
 or returned by the body function.  
`@Lid{lua_resume}`  
 returns  
`@Lid{LUA_YIELD}` if the coroutine yields,  
`@Lid{LUA_OK}` if the coroutine finishes its execution  
 without errors,  
 or an error code in case of errors `@see{statuscodes}`.  
 In case of errors,  
 the error object is on the top of the stack.

To resume a coroutine,  
 you remove the `@id{*nresults}` yielded values from its stack,  
 push the values to be passed as results from `@id{yield}`,  
 and then call `@Lid{lua_resume}`.

The parameter `@id{from}` represents the coroutine that is resuming `@id{L}`.  
 If there is no such coroutine,  
 this parameter can be `@id{NULL}`.

}

`@APIEntry{void lua_rotate (lua_State *L, int idx, int n);|`  
`@apii{0,0,-}`

Rotates the stack elements between the valid index `@id{idx}`  
 and the top of the stack.  
 The elements are rotated `@id{n}` positions in the direction of the top,  
 for a positive `@id{n}`,  
 or `@T{-n}` positions in the direction of the bottom,  
 for a negative `@id{n}`.  
 The absolute value of `@id{n}` must not be greater than the size  
 of the slice being rotated.  
 This function  
 cannot be called with a pseudo-index,  
 because a pseudo-index is not an actual stack position.

}

`@APIEntry{void lua_setallocf (lua_State *L, lua_Alloc f, void *ud);|`  
`@apii{0,0,-}`

Changes the `@x{allocator function}` of a given state to `@id{f}` with user data `@id{ud}`.

}

```
@APIEntry{void lua_setfield (lua_State *L, int index, const char *k);|
@apii{1,0,e}
```

Does the equivalent to `@T{t[k] = v}`,  
where `@id{t}` is the value at the given index  
and `@id{v}` is the value on the top of the stack.

This function pops the value from the stack.  
As in Lua, this function may trigger a metamethod  
for the `@Q{newindex}` event `@see{metatable}`.

}

```
@APIEntry{void lua_setglobal (lua_State *L, const char *name);|
@apii{1,0,e}
```

Pops a value from the stack and  
sets it as the new value of global `@id{name}`.

}

```
@APIEntry{void lua_seti (lua_State *L, int index, lua_Integer n);|
@apii{1,0,e}
```

Does the equivalent to `@T{t[n] = v}`,  
where `@id{t}` is the value at the given index  
and `@id{v}` is the value on the top of  
the stack.

This function pops the value from the stack.  
As in Lua, this function may trigger a metamethod  
for the `@Q{newindex}` event `@see{metatable}`.

}

```
@APIEntry{int lua_setiuservalue (lua_State *L, int index, int n);|
@apii{1,0,-}
```

Pops a value from the stack and sets it as  
the new `@id{n}`-th user value associated to the  
full userdata at the given index.  
Returns 0 if the userdata does not have that value.



```
}
```

```
@APIEntry{int lua_setmetatable (lua_State *L, int index);|  
@apii{1,0,-}}
```

Pops a table or @nil from the stack and  
sets that value as the new metatable for the value at the given index.  
(@nil means no metatable.)

(For historical reasons, this function returns an @id{int},  
which now is always 1.)

```
}
```

```
@APIEntry{void lua_settable (lua_State *L, int index);|  
@apii{2,0,e}}
```

Does the equivalent to @T{t[k] = v},  
where @id{t} is the value at the given index,  
@id{v} is the value on the top of the stack,  
and @id{k} is the value just below the top.

This function pops both the key and the value  
from the stack.  
As in Lua, this function may trigger a metamethod  
for the @Q{newindex} event @see{metatable}.

```
}
```

```
@APIEntry{void lua_settop (lua_State *L, int index);|  
@apii{?,?,e}}
```

Accepts any index, @N{or 0},  
and sets the stack top to this index.  
If the new top is greater than the old one,  
then the new elements are filled with @nil.  
If @id{index} @N{is 0}, then all stack elements are removed.

This function can run arbitrary code when removing an index  
marked as to-be-closed from the stack.

```
}
```

```
@APIEntry{void lua_setwarnf (lua_State *L, lua_WarnFunction f, void *ud);|  
@apii{0,0,-}}
```

Sets the @x{warning function} to be used by Lua to emit warnings

@see{lua\_WarnFunction}.

The @id{ud} parameter sets the value @id{ud} passed to the warning function.

}

@APIEntry{typedef struct lua\_State lua\_State;|

An opaque structure that points to a thread and indirectly (through the thread) to the whole state of a Lua interpreter.

The Lua library is fully reentrant:

it has no global variables.

All

information about a state is accessible through this structure.

A pointer to this structure must be passed as the first argument to every function in the library, except to @Lid{lua\_newstate}, which creates a Lua state from scratch.

}

@APIEntry{int lua\_status (lua\_State \*L);|

@apii{0,0,-}

Returns the status of the thread @id{L}.

The status can be @Lid{LUA\_OK} for a normal thread,

an error code if the thread finished the execution

of a @Lid{lua\_resume} with an error,

or @Lid{LUA\_YIELD} if the thread is suspended.

You can call functions only in threads with status @Lid{LUA\_OK}.

You can resume threads with status @Lid{LUA\_OK}

(to start a new coroutine) or @Lid{LUA\_YIELD}

(to resume a coroutine).

}

@APIEntry{size\_t lua\_stringtonumber (lua\_State \*L, const char \*s);|

@apii{0,1,-}

Converts the zero-terminated string @id{s} to a number,

pushes that number into the stack,

and returns the total size of the string,

that is, its length plus one.

The conversion can result in an integer

or a float,

according to the lexical conventions of Lua @see{lexical}.

The string may have leading and trailing whitespaces and a sign.  
If the string is not a valid numeral,  
returns 0 and pushes nothing.  
(Note that the result can be used as a boolean,  
true if the conversion succeeds.)

}

```
@APIEntry{int lua_toboolean (lua_State *L, int index);|  
@apii{0,0,-}}
```

Converts the Lua value at the given index to a @N{C boolean}  
value (@N{0 or 1}).

Like all tests in Lua,

@Lid{lua\_toboolean} returns true for any Lua value  
different from @false and @nil;  
otherwise it returns false.

(If you want to accept only actual boolean values,  
use @Lid{lua\_isboolean} to test the value's type.)

}

```
@APIEntry{lua_CFunction lua_tocfunction (lua_State *L, int index);|  
@apii{0,0,-}}
```

Converts a value at the given index to a @N{C function}.  
That value must be a @N{C function};  
otherwise, returns @id{NULL}.

}

```
@APIEntry{void lua_toclose (lua_State *L, int index);|  
@apii{0,0,v}}
```

Marks the given index in the stack as a  
to-be-closed

slot @see{to-be-closed}.

Like a to-be-closed variable in Lua,  
the value at that slot in the stack will be closed  
when it goes out of scope.

Here, in the context of a C function,  
to go out of scope means that the running function returns to Lua,  
or there is an error,

or the slot is removed from the stack through

@Lid{lua\_settop} or @Lid{lua\_pop},  
or there is a call to @Lid{lua\_closeslot}.

A slot marked as to-be-closed should not be removed from the stack  
by any other function in the API except @Lid{lua\_settop} or @Lid{lua\_pop},

unless previously deactivated by @Lid{lua\_closeslot}.

This function raises an error if the value at the given slot  
neither has a @idx{\_\_close} metamethod nor is a false value.

This function should not be called for an index  
that is equal to or below an active to-be-closed slot.

Note that, both in case of errors and of a regular return,  
by the time the @idx{\_\_close} metamethod runs,  
the @N{C stack} was already unwound,  
so that any automatic @N{C variable}  
declared in the calling function  
(e.g., a buffer) will be out of scope.

}

@APIEntry{lua\_Integer lua\_tointeger (lua\_State \*L, int index);|  
@apii{0,0,-}}

Equivalent to @Lid{lua\_tointegerx} with @id{isnum} equal to @id{NULL}.

}

@APIEntry{lua\_Integer lua\_tointegerx (lua\_State \*L, int index, int \*isnum);|  
@apii{0,0,-}}

Converts the Lua value at the given index  
to the signed integral type @Lid{lua\_Integer}.  
The Lua value must be an integer,  
or a number or string convertible to an integer @see{coercion};  
otherwise, @id{lua\_tointegerx} @N{returns 0}.

If @id{isnum} is not @id{NULL},  
its referent is assigned a boolean value that  
indicates whether the operation succeeded.

}

@APIEntry{const char \*lua\_tolstring (lua\_State \*L, int index, size\_t \*len);|  
@apii{0,0,m}}

Converts the Lua value at the given index to a @N{C string}.  
If @id{len} is not @id{NULL},  
it sets @T{\*len} with the string length.  
The Lua value must be a string or a number;  
otherwise, the function returns @id{NULL}.  
If

the value is a number,  
then @id{lua\_tolstring} also  
@emph{changes the actual value in the stack to a string}.  
(This change confuses @Lid{lua\_next}  
when @id{lua\_tolstring} is applied to keys during a table traversal.)

@id{lua\_tolstring} returns a pointer  
to a string inside the Lua state @see{constchar}.  
This string always has a zero (@Char{\0})  
after its last character (as @N{in C}),  
but can contain other zeros in its body.

This function can raise memory errors only  
when converting a number to a string  
(as then it may create a new string).

}

@APIEntry{lua\_Number lua\_tonumber (lua\_State \*L, int index);|  
@apii{0,0,-}}

Equivalent to @Lid{lua\_tonumberx} with @id{isnum} equal to @id{NULL}.

}

@APIEntry{lua\_Number lua\_tonumberx (lua\_State \*L, int index, int \*isnum);|  
@apii{0,0,-}}

Converts the Lua value at the given index  
to the @N{C type} @Lid{lua\_Number} @seeC{lua\_Number}.  
The Lua value must be a number or a string convertible to a number  
@see{coercion};  
otherwise, @Lid{lua\_tonumberx}  
@N{returns 0}.

If @id{isnum} is not @id{NULL},  
its referent is assigned a boolean value that  
indicates whether the operation succeeded.

}

@APIEntry{const void \*lua\_topointer (lua\_State \*L, int index);|  
@apii{0,0,-}}

Converts the value at the given index to a generic  
@N{C pointer} (@T{void\*}).  
The value can be a userdata, a table, a thread, a string, or a function;  
otherwise, @id{lua\_topointer} returns @id{NULL}.

Different objects will give different pointers.

There is no way to convert the pointer back to its original value.

Typically this function is used only for hashing and debug information.

```
}
```

```
@APIEntry{const char *lua_tolstring (lua_State *L, int index);|
@apii{0,0,m}
```

Equivalent to @Lid{lua\_tolstring} with @id{len} equal to @id{NULL}.

```
}
```

```
@APIEntry{lua_State *lua_tothread (lua_State *L, int index);|
@apii{0,0,-}
```

Converts the value at the given index to a Lua thread  
(represented as @T{lua\_State\*}).

This value must be a thread;

otherwise, the function returns @id{NULL}.

```
}
```

```
@APIEntry{void
*lua_touserdata (lua_State *L, int index);|
@apii{0,0,-}
```

If the value at the given index is a full userdata,  
returns its memory-block address.

If the value is a light userdata,  
returns its value (a pointer).

Otherwise, returns @id{NULL}.

```
}
```

```
@APIEntry{int lua_type (lua_State *L, int index);|
@apii{0,0,-}
```

Returns the type of the value in the given valid index,  
or @id{LUA\_TNONE} for a non-valid but acceptable index.

The types returned by @Lid{lua\_type} are coded by the following constants  
defined in @id{lua.h}:

```
@defid{LUA_TNIL},
@defid{LUA_TNUMBER},
@defid{LUA_TBOOLEAN},
@defid{LUA_TSTRING},
@defid{LUA_TTABLE},
```

```
@defid{LUA_TFUNCTION},
@defid{LUA_TUSERDATA},
@defid{LUA_TTHREAD},
and
@defid{LUA_TLIGHTUSERDATA}.
```

```
}
```

```
@APIEntry{const char *lua_typename (lua_State *L, int tp);|
@apii{0,0,-}}
```

Returns the name of the type encoded by the value @id{tp}, which must be one the values returned by @Lid{lua\_type}.

```
}
```

```
@APIEntry{typedef @ldots lua_Unsigned;|
```

The unsigned version of @Lid{lua\_Integer}.

```
}
```

```
@APIEntry{int
lua_upvalueindex (int i);|
@apii{0,0,-}}
```

Returns the pseudo-index that represents the @id{i}-th upvalue of the running function @see{c-closure}.  
@id{i} must be in the range @M{[1,256]}.

```
}
```

```
@APIEntry{lua_Number lua_version (lua_State *L);|
@apii{0,0,-}}
```

Returns the version number of this core.

```
}
```

```
@APIEntry{
typedef void (*lua_WarnFunction) (void *ud, const char *msg, int tocont);|
```

The type of @x{warning function}s, called by Lua to emit warnings.

The first parameter is an opaque pointer set by @Lid{lua\_setwarnf}.

The second parameter is the warning message.

The third parameter is a boolean that indicates whether the message is

to be continued by the message in the next call.

See @Lid{warn} for more details about warnings.

}

```
@APIEntry{
void lua_warning (lua_State *L, const char *msg, int tocont);|
@apii{0,0,-}
```

Emits a warning with the given message.

A message in a call with @id{tocont} true should be continued in another call to this function.

See

@Lid{warn} for more details about warnings.

}

```
@APIEntry{
typedef int (*lua_Writer) (lua_State *L,
                           const void* p,
                           size_t sz,
                           void* ud);|
```

The type of the writer function used by @Lid{lua\_dump}.  
Every time @Lid{lua\_dump} produces another piece of chunk,  
it calls the writer,  
passing along the buffer to be written (@id{p}),  
its size (@id{sz}),  
and the @id{ud} parameter supplied to @Lid{lua\_dump}.

The writer returns an error code:

@N{0 means} no errors;  
any other value means an error and stops @Lid{lua\_dump} from  
calling the writer again.

}

```
@APIEntry{void lua_xmove (lua_State *from, lua_State *to, int n);|
@apii{?,?,-}
```

Exchange values between different threads of the same state.

This function pops @id{n} values from the stack @id{from},  
and pushes them onto the stack @id{to}.

}



```
@APIEntry{int lua_yield (lua_State *L, int nresults);|
@apii{?,?,v}
```

This function is equivalent to @Lid{lua\_yieldk},  
but it has  
no continuation @see{continuations}.  
Therefore, when the thread resumes,  
it continues the function that called  
the function calling @id{lua\_yield}.  
To avoid surprises,  
this function should be called only in a tail call.

```
}
```

```
@APIEntry{
int lua_yieldk (lua_State *L,
               int nresults,
               lua_KContext ctx,
               lua_KFunction k);|
@apii{?,?,v}
```

Yields a coroutine (thread).

When a @N{C function} calls @Lid{lua\_yieldk},  
the running coroutine suspends its execution,  
and the call to @Lid{lua\_resume} that started this coroutine returns.  
The parameter @id{nresults} is the number of values from the stack  
that will be passed as results to @Lid{lua\_resume}.

When the coroutine is resumed again,  
Lua calls the given @x{continuation function} @id{k} to continue  
the execution of the @N{C function} that yielded @see{continuations}.  
This continuation function receives the same stack  
from the previous function,  
with the @id{n} results removed and  
replaced by the arguments  
passed to @Lid{lua\_resume}.  
Moreover,  
the continuation function receives the value @id{ctx}  
that was passed to @Lid{lua\_yieldk}.

Usually, this function does not return;  
when the coroutine eventually resumes,  
it continues executing the continuation function.  
However, there is one special case,  
which is when this function is called

from inside a line or a count hook `@see{debugI}`.  
 In that case, `@id{lua_yieldk}` should be called with no continuation  
 (probably in the form of `@Lid{lua_yield}`) and no results,  
 and the hook should return immediately after the call.  
 Lua will yield and,  
 when the coroutine resumes again,  
 it will continue the normal execution  
 of the (Lua) function that triggered the hook.

This function can raise an error if it is called from a thread  
 with a pending C call with no continuation function  
 (what is called a `@emphx{C-call boundary}`),  
 or it is called from a thread that is not running inside a resume  
 (typically the main thread).

}

}

`@sect2{debugI @title{The Debug  
 Interface}`

Lua has no built-in debugging facilities.  
 Instead, it offers a special interface  
 by means of functions and `@emph{hooks}`.  
 This interface allows the construction of different  
 kinds of debuggers, profilers, and other tools  
 that need `@Q{inside information}` from the interpreter.

```
@APIEntry{
typedef struct lua_Debug {
  int event;
  const char *name;      /* (n) */
  const char *namewhat;  /* (n) */
  const char *what;      /* (S) */
  const char *source;    /* (S) */
  size_t srclen;         /* (S) */
  int currentline;       /* (l) */
  int linedefined;       /* (S) */
  int lastlinedefined;   /* (S) */
  unsigned char nups;     /* (u) number of upvalues */
  unsigned char nparams; /* (u) number of parameters */
  char isvararg;         /* (u) */
  char istailcall;       /* (t) */
  unsigned short ftransfer; /* (r) index of first value transferred */
  unsigned short ntransfer; /* (r) number of transferred values
```

```

*/
char short_src[LUA_IDSIZE]; /* (S) */
/* private part */
@rep{other fields}
} lua_Debug;
|

```

A structure used to carry different pieces of information about a function or an activation record.

@Lid{lua\_getstack} fills only the private part of this structure, for later use.

To fill the other fields of @Lid{lua\_Debug} with useful information, you must call @Lid{lua\_getinfo} with an appropriate parameter.

(Specifically, to get a field,

you must add the letter between parentheses in the field's comment to the parameter @id{what} of @Lid{lua\_getinfo}.)

The fields of @Lid{lua\_Debug} have the following meaning:

@description{

@item{@id{source}|

the source of the chunk that created the function.

If @T{source} starts with a @Char{@At},

it means that the function was defined in a file where

the file name follows the @Char{@At}.

If @T{source} starts with a @Char{=},

the remainder of its contents describes the source in a user-dependent manner.

Otherwise,

the function was defined in a string

where

@T{source} is that string.

}

@item{@id{srclen}|

The length of the string @id{source}.

}

@item{@id{short\_src}|

a @Q{printable} version of @T{source}, to be used in error messages.

}

@item{@id{linedefined}|

the line number where the definition of the function starts.

}

@item{@id{lastlinedefined}|

the line number where the definition of the function ends.

}

@item{@id{what}|  
the string @T{"Lua"} if the function is a Lua function,  
@T{"C"} if it is a @N{C function},  
@T{"main"} if it is the main part of a chunk.  
}

@item{@id{currentline}|  
the current line where the given function is executing.  
When no line information is available,  
@T{currentline} is set to @num{-1}.  
}

@item{@id{name}|  
a reasonable name for the given function.  
Because functions in Lua are first-class values,  
they do not have a fixed name:  
some functions can be the value of multiple global variables,  
while others can be stored only in a table field.  
The @T{lua\_getinfo} function checks how the function was  
called  
to find a suitable name.  
If it cannot find a name,  
then @id{name} is set to @id{NULL}.  
}

@item{@id{namewhat}|  
explains the @T{name} field.  
The value of @T{namewhat} can be  
@T{"global"}, @T{"local"}, @T{"method"},  
@T{"field"}, @T{"upvalue"}, or @T{""} (the empty string),  
according to how the function was called.  
(Lua uses the empty string when no other option seems to apply.)  
}

@item{@id{istailcall}|  
true if this function invocation was called by a tail call.  
In this case, the caller of this level is not in the stack.  
}

@item{@id{nups}|  
the number of upvalues of the function.  
}

@item{@id{nparams}|  
the number of parameters of the function  
(always @N{0} for @N{C functions}).  
}

@item{@id{isvararg}}|  
true if the function is a variadic function  
(always true for @N{C functions}).  
}

@item{@id{ftransfer}}|  
the index in the stack of the first value being @Q{transferred},  
that is, parameters in a call or return values in a return.  
(The other values are in consecutive indices.)  
Using this  
index, you can access and modify these values  
through @Lid{lua\_getlocal} and @Lid{lua\_setlocal}.  
This field is only meaningful during a  
call hook, denoting the first parameter,  
or a return hook, denoting the first value being returned.  
(For call hooks, this value is always 1.)  
}

@item{@id{ntransfer}}|  
The number of values being transferred (see previous item).  
(For calls of Lua functions,  
this value is always equal to @id{nparams}.)  
}

}

}

@APIEntry{lua\_Hook lua\_gethook (lua\_State \*L);|  
@apii{0,0,-}}

Returns the current hook function.

}

@APIEntry{int lua\_gethookcount (lua\_State \*L);|  
@apii{0,0,-}}

Returns the current hook count.

}

@APIEntry{int lua\_gethookmask (lua\_State \*L);|  
@apii{0,0,-}}

Returns the current hook mask.

```
}
```

```
@APIEntry{int lua_getinfo (lua_State *L, const char *what, lua_Debug *ar);|
@apii{0|1,0|1|2,m}
```

Gets information about a specific function or function invocation.

To get information about a function invocation,  
the parameter `@id{ar}` must be a valid activation  
record that was  
filled by a previous call to `@Lid{lua_getstack}` or  
given as argument to a hook `@seeC{lua_Hook}`.

To get information about a function, you push it onto the stack  
and start the `@id{what}` string with the character `@Char{>}`.

(In that case,

`@id{lua_getinfo}` pops the function from the top of the stack.)

For instance, to know in which line a function `@id{f}` was defined,  
you can write the following code:

```
@verbatim{
lua_Debug ar;
lua_getglobal(L, "f"); /* get global 'f' */
lua_getinfo(L, ">S", &ar);
printf("%d\n", ar.linedefined);
}
```

Each character in the string `@id{what}`  
selects some fields of the structure `@id{ar}` to be filled or  
a value to be pushed on the stack.  
(These characters are also documented in the declaration of  
the structure `@Lid{lua_Debug}`,  
between parentheses in the comments following each field.)  
`@description{`

```
@item{@Char{f}|
pushes onto the stack the function that is
running at the given level;
}
```

```
@item{@Char{l}| fills in the field @id{currentline};
}
```

```
@item{@Char{n}|
fills in the fields @id{name} and @id{namewhat};
}
```

```
@item{@Char{r}| fills in the fields @id{ftransfer} and @id{ntransfer};
}
```

```
@item{@Char{S}|
fills in the fields @id{source}, @id{short_src},
@id{linedefined}, @id{lastlinedefined}, and @id{what};
}
```

```
@item{@Char{t}| fills in the field @id{istailcall};
}
```

```
@item{@Char{u}| fills in the fields
@id{nups}, @id{nparams}, and @id{isvararg};
}
```

```
@item{@Char{L}|
pushes onto the stack a table whose indices are
the lines on the function with some associated code,
that is, the lines where you can put a break point.
(Lines with no code include empty lines and comments.)
If this option is given together with option @Char{f},
its table is pushed after the function.
This is the only option that can raise a memory error.
}
```

```
}
```

This function returns 0 to signal an invalid option in @id{what};  
even then the valid options are handled correctly.

```
}
```

```
@APIEntry{const char *lua_getlocal (lua_State *L, const lua_Debug *ar, int n);|
@apii{0,0|1,-}}
```

**Gets**  
information about a local variable or a temporary value  
of a given activation record or a given function.

In the first case,  
the parameter @id{ar} must be a valid activation record that was  
filled by a previous call to @Lid{lua\_getstack} or  
given as argument to a hook @seeC{lua\_Hook}.  
The index @id{n} selects which local variable to inspect;  
see @Lid{debug.getlocal} for details about variable indices  
and names.

@Lid{lua\_getlocal} pushes the variable's value onto the stack  
and returns its name.

In the second case, @id{ar} must be @id{NULL} and the function to be inspected must be on the top of the stack.

In this case, only parameters of Lua functions are visible (as there is no information about what variables are active) and no values are pushed onto the stack.

Returns @id{NULL} (and pushes nothing) when the index is greater than the number of active local variables.

}

```
@APIEntry{int lua_getstack (lua_State *L, int level, lua_Debug *ar);|
@apii{0,0,-}}
```

Gets information about the interpreter runtime stack.

This function fills parts of a @Lid{lua\_Debug} structure with an identification of the @emph{activation record} of the function executing at a given level. @N{Level 0} is the current running function, whereas level @M{n+1} is the function that has called level @M{n} (except for tail calls, which do not count in the stack). When called with a level greater than the stack depth, @Lid{lua\_getstack} returns 0; otherwise it returns 1.

}

```
@APIEntry{const char *lua_getupvalue (lua_State *L, int funcindex, int n);|
@apii{0,0|1,-}}
```

Gets information about the @id{n}-th upvalue of the closure at index @id{funcindex}.

It pushes the upvalue's value onto the stack and returns its name.

Returns @id{NULL} (and pushes nothing) when the index @id{n} is greater than the number of upvalues.

See @Lid{debug.getupvalue} for more information about upvalues.

}

```
@APIEntry{typedef void (*lua_Hook) (lua_State *L, lua_Debug *ar);|
```

Type for debugging hook functions.



Whenever a hook is called, its `@id{ar}` argument has its field `@id{event}` set to the specific event that triggered the hook. Lua identifies these events with the following constants: `@defid{LUA_HOOKCALL}`, `@defid{LUA_HOOKRET}`, `@defid{LUA_HOOKTAILCALL}`, `@defid{LUA_HOOKLINE}`, and `@defid{LUA_HOOKCOUNT}`. Moreover, for line events, the field `@id{currentline}` is also set. To get the value of any other field in `@id{ar}`, the hook must call `@Lid{lua_getinfo}`.

For call events, `@id{event}` can be `@id{LUA_HOOKCALL}`, the normal value, or `@id{LUA_HOOKTAILCALL}`, for a tail call; in this case, there will be no corresponding return event.

While Lua is running a hook, it disables other calls to hooks. Therefore, if a hook calls back Lua to execute a function or a chunk, this execution occurs without any calls to hooks.

Hook functions cannot have continuations, that is, they cannot call `@Lid{lua_yieldk}`, `@Lid{lua_pcallk}`, or `@Lid{lua_callk}` with a non-null `@id{k}`.

Hook functions can yield under the following conditions:

Only count and line events can yield; to yield, a hook function must finish its execution calling `@Lid{lua_yield}` with `@id{nresults}` equal to zero (that is, with no values).

}

```
@APIEntry{void lua_sethook (lua_State *L, lua_Hook f, int mask, int count);|
@apii{0,0,-}
```

Sets the debugging hook function.

Argument `@id{f}` is the hook function. `@id{mask}` specifies on which events the hook will be called: it is formed by a bitwise OR of the constants `@defid{LUA_MASKCALL}`, `@defid{LUA_MASKRET}`, `@defid{LUA_MASKLINE}`, and `@defid{LUA_MASKCOUNT}`. The `@id{count}` argument is only meaningful when the mask includes `@id{LUA_MASKCOUNT}`.

For each event, the hook is called as explained below:

@description{

@item{The call hook| is called when the interpreter calls a function.  
The hook is called just after Lua enters the new function.  
}

@item{The return hook| is called when the interpreter returns from a function.  
The hook is called just before Lua leaves the function.  
}

@item{The line hook| is called  
when the interpreter is about to  
start the execution of a new line of code,  
or when it jumps back in the code (even to the same line).  
This event only happens while Lua is executing a Lua function.  
}

@item{The count hook| is called after the interpreter executes every  
@T{count} instructions.  
This event only happens while Lua is executing a Lua function.  
}  
  
}

Hooks are disabled by setting @id{mask} to zero.  
  
}

@APIEntry{const char \*lua\_setlocal (lua\_State \*L, const lua\_Debug \*ar, int n);|  
@apii{0|1,0,-}}

Sets the value of a local variable of a given activation record.  
It assigns the value on the top of the stack  
to the variable and returns its name.  
It also pops the value from the stack.

Returns @id{NULL} (and pops nothing)  
when the index is greater than  
the number of active local variables.

Parameters @id{ar} and @id{n} are as in the function @Lid{lua\_getlocal}.  
  
}

@APIEntry{const char \*lua\_setupvalue (lua\_State \*L, int funcindex, int n);|  
@apii{0|1,0,-}}

Sets the value of a closure's

upvalue.

It assigns the value on the top of the stack  
to the upvalue and returns its name.

It also pops the value from the stack.

Returns @id{NULL} (and pops nothing)  
when the index @id{n} is greater than the number of upvalues.

Parameters @id{funcindex} and @id{n} are as in  
the function @Lid{lua\_getupvalue}.

}

```
@APIEntry{void *lua_upvalueid (lua_State *L, int funcindex, int n);|
@apii{0,0,-}
```

Returns a unique identifier for the upvalue numbered @id{n}  
from the closure at index @id{funcindex}.

These unique identifiers allow a program to check whether different  
closures share upvalues.

Lua closures that share an upvalue  
(that is, that access a same external local variable)  
will return identical ids for those upvalue indices.

Parameters @id{funcindex} and @id{n} are as in  
the function @Lid{lua\_getupvalue},  
but @id{n} cannot be greater than the number of upvalues.

}

```
@APIEntry{
void lua_upvaluejoin (lua_State *L, int funcindex1, int n1,
                    int
                    funcindex2, int n2);|
@apii{0,0,-}
```

Make the @id{n1}-th upvalue of the Lua closure at index @id{funcindex1}  
refer to the @id{n2}-th upvalue of the Lua closure at index @id{funcindex2}.

}

}

}

```
@C{-----}
```

@sect1{auxlib|@title{The Auxiliary Library}}

@simplesect{

@index{luauxlib.h}

The @def{auxiliary library} provides several convenient functions to interface C with Lua.

While the basic API provides the primitive functions for all interactions between C and Lua, the auxiliary library provides higher-level functions for some common tasks.

All functions and types from the auxiliary library are defined in header file @id{luauxlib.h} and have a prefix @id{luaL\_}.

All functions in the auxiliary library are built on top of the basic API, and so they provide nothing that cannot be done with that API. Nevertheless, the use of the auxiliary library ensures more consistency to your code.

Several functions in the auxiliary library use internally some extra stack slots. When a function in the auxiliary library uses less than five slots, it does not check the stack size; it simply assumes that there are enough slots.

Several functions in the auxiliary library are used to check @N{C function} arguments. Because the error message is formatted for arguments (e.g., @St{bad argument #1}), you should not use these functions for other stack values.

Functions called @id{luaL\_check\*} always raise an error if the check is not satisfied.

}

@sect2{@title{Functions and Types}}

Here we list all functions and types from the auxiliary library in alphabetical order.

```
@APIEntry{void luaL_addchar (luaL_Buffer *B, char c);|
@apii{?,?,m}
```

Adds the byte @id{c} to the buffer @id{B}  
@seeC{luaL\_Buffer}.

```
}
```

```
@APIEntry{
const void luaL_addgsub (luaL_Buffer *B, const char *s,
                        const char *p, const char *r);|
@apii{?,?,m}
```

Adds a copy of the string @id{s} to the buffer @id{B} @seeC{luaL\_Buffer},  
replacing any  
occurrence of the string @id{p}  
with the string @id{r}.

```
}
```

```
@APIEntry{void luaL_addlstring (luaL_Buffer *B, const char *s, size_t l);|
@apii{?,?,m}
```

Adds the string pointed to by @id{s} with length @id{l} to  
the buffer @id{B}  
@seeC{luaL\_Buffer}.

The string can contain @x{embedded zeros}.

```
}
```

```
@APIEntry{void luaL_addsize (luaL_Buffer *B, size_t n);|
@apii{?,?,-}
```

Adds to the buffer @id{B}  
a string of length @id{n} previously copied to the  
buffer area @seeC{luaL\_prepbuffer}.

```
}
```

```
@APIEntry{void luaL_addstring (luaL_Buffer *B, const char *s);|
@apii{?,?,m}
```

Adds the zero-terminated string pointed to by @id{s}  
to the buffer @id{B}  
@seeC{luaL\_Buffer}.

```
}
```

```
@APIEntry{void luaL_addvalue (luaL_Buffer *B);|
@apii{?,?,m}
```

Adds the value on the top of the stack  
to the buffer @id{B}  
@seeC{luaL\_Buffer}.  
Pops the value.

This is the only function on string buffers that can (and must)  
be called with an extra element on the stack,  
which is the value to be added to the buffer.

```
}
```

```
@APIEntry{
void luaL_argcheck
(lua_State *L,
    int cond,
    int arg,
    const char *extramsg);|
@apii{0,0,v}
```

Checks whether @id{cond} is true.  
If it is not, raises an error with a standard message @seeF{luaL\_argerror}.

```
}
```

```
@APIEntry{int luaL_argerror (lua_State *L, int arg, const char *extramsg);|
@apii{0,0,v}
```

Raises an error reporting a problem with argument @id{arg}  
of the @N{C function} that called it,  
using a standard message  
that includes @id{extramsg} as a comment:  
@verbatim{  
bad argument #@rep{arg} to '@rep{funcname}' (@rep{extramsg})  
}  
This function never returns.

```
}
```

```
@APIEntry{
void luaL_argexpected (lua_State *L,
    int cond,
    int arg,
    const char *tname);|
@apii{0,0,v}
```

Checks whether @id{cond} is true.

If it is not, raises an error about the type of the argument @id{arg} with a standard message @seeF{luaL\_typeerror}.

}

@APIEntry{typedef struct luaL\_Buffer luaL\_Buffer;|

Type for

a @def{string buffer}.

A string buffer allows @N{C code} to build Lua strings piecemeal.

Its pattern of use is as follows:

@itemize{

@item{First declare a variable @id{b} of type @Lid{luaL\_Buffer}.)}

@item{Then initialize it with a call @T{luaL\_buffinit(L, &b)}.)}

@item{

Then add string pieces to the buffer calling any of the @id{luaL\_add\*} functions.

}

@item{

Finish by calling @T{luaL\_pushresult(&b)}.

This call leaves the final string on the top of the stack.

}

}

If you know beforehand the maximum size of the resulting string, you can use the buffer like this:

@itemize{

@item{First declare a variable @id{b} of type @Lid{luaL\_Buffer}.)}

@item{Then initialize it and preallocate a space of size @id{sz} with a call @T{luaL\_buffinitsize(L, &b, sz)}.)}

@item{Then produce the string into that space.}

@item{

Finish by calling @T{luaL\_pushresultsize(&b, sz)}, where @id{sz} is the total size of the resulting string copied into that space (which may be less than or equal to the preallocated

```

size).
}

}

```

During its normal operation,  
a string buffer uses a variable number of stack slots.  
So, while using a buffer, you cannot assume that you know where  
the top of the stack is.  
You can use the stack between successive calls to buffer operations  
as long as that use is balanced;  
that is,  
when you call a buffer operation,  
the stack is at the same level  
it was immediately after the previous buffer operation.  
(The only exception to this rule is @Lid{luaL\_addvalue}.)  
After calling @Lid{luaL\_pushresult},  
the stack is back to its level when the buffer was initialized,  
plus the final string on its top.

```

}

```

```

@APIEntry{char *luaL_buffaddr (luaL_Buffer *B);|
@apii{0,0,-}

```

Returns the address of the current content of buffer @id{B}  
@seeC{luaL\_Buffer}.  
Note that any addition to the buffer may invalidate this address.

```

}

```

```

@APIEntry{void luaL_buffinit (lua_State *L, luaL_Buffer *B);|
@apii{0,?, -}

```

Initializes a buffer @id{B}  
@seeC{luaL\_Buffer}.  
This function does not allocate any space;  
the  
buffer must be declared as a variable.

```

}

```

```

@APIEntry{size_t luaL_bufflen (luaL_Buffer *B);|
@apii{0,0,-}

```

Returns the length of the current content of buffer @id{B}  
@seeC{luaL\_Buffer}.



}

```
@APIEntry{char *luaL_buffinitsize (lua_State *L, luaL_Buffer *B, size_t sz);|
@apii{?,?,m}
```

Equivalent to the sequence

```
@Lid{luaL_buffinit}, @Lid{luaL_prepbuffsize}.
```

}

```
@APIEntry{void luaL_buffsub (luaL_Buffer *B, int n);|
@apii{?,?,-}
```

Removes @id{n} bytes from the buffer @id{B}

```
@seeC{luaL_Buffer}.
```

The buffer must have at least that many bytes.

}

```
@APIEntry{int luaL_callmeta (lua_State *L, int obj, const char *e);|
@apii{0,0|1,e}
```

Calls a metamethod.

If the object at index @id{obj} has a metatable and this

metatable has a field @id{e},

this function calls this field passing the object as its only argument.

In this case this function returns true and pushes onto the stack the value returned by the call.

If there is no metatable or no metamethod,

this function returns false without pushing any value on the stack.

}

```
@APIEntry{void luaL_checkany (lua_State *L, int arg);|
@apii{0,0,v}
```

Checks whether the function has an argument

of any type (including @nil) at position @id{arg}.

}

```
@APIEntry{lua_Integer luaL_checkinteger (lua_State *L, int arg);|
@apii{0,0,v}
```

Checks whether the function argument @id{arg} is an integer

(or can be converted to an integer)

and returns this integer.

```
}
```

```
@APIEntry{const char *luaL_checklstring (lua_State *L, int arg, size_t *l);|
@apii{0,0,v}
```

Checks whether the function argument @id{arg} is a string  
and returns this string;  
if @id{l} is not @id{NULL} fills its referent  
with the string's length.

This function uses @Lid{lua\_tolstring} to get its result,  
so all conversions and caveats of that function apply here.

```
}
```

```
@APIEntry{lua_Number luaL_checknumber (lua_State *L, int arg);|
@apii{0,0,v}
```

Checks whether the function argument @id{arg} is a number  
and returns this number converted to a @id{lua\_Number}.

```
}
```

```
@APIEntry{
int luaL_checkoption (lua_State
*L,
    int arg,
    const char *def,
    const char *const lst[]);|
@apii{0,0,v}
```

Checks whether the function argument @id{arg} is a string and  
searches for this string in the array @id{lst}  
(which must be NULL-terminated).  
Returns the index in the array where the string was found.  
Raises an error if the argument is not a string or  
if the string cannot be found.

If @id{def} is not @id{NULL},  
the function uses @id{def} as a default value when  
there is no argument @id{arg} or when this argument is @nil.

This is a useful function for mapping strings to @N{C enums}.  
(The usual convention in Lua libraries is  
to use strings instead of numbers to select options.)

}

```
@APIEntry{void luaL_checkstack (lua_State *L, int sz, const char *msg);|
@apii{0,0,v}
```

Grows the stack size to @T{top + sz} elements,  
raising an error if the stack cannot grow to that size.  
@id{msg} is an additional text to go into the error message  
(or @id{NULL} for no  
additional text).

}

```
@APIEntry{const char *luaL_checkstring (lua_State *L, int arg);|
@apii{0,0,v}
```

Checks whether the function argument @id{arg} is a string  
and returns this string.

This function uses @Lid{lua\_tolstring} to get its result,  
so all conversions and caveats of that function apply here.

}

```
@APIEntry{void luaL_checktype (lua_State *L, int arg, int t);|
@apii{0,0,v}
```

Checks whether the function argument @id{arg} has type @id{t}.  
See @Lid{lua\_type} for the encoding of types for @id{t}.

}

```
@APIEntry{void *luaL_checkudata (lua_State *L, int arg, const char *tname);|
@apii{0,0,v}
```

Checks whether the function argument @id{arg} is a userdata  
of the type @id{tname} @seeC{luaL\_newmetatable} and  
returns the userdata's memory-block address @seeC{lua\_touserdata}.

}

```
@APIEntry{void luaL_checkversion (lua_State *L);|
@apii{0,0,v}
```

Checks whether the code making the call and the Lua library being called  
are using the same version of Lua and the same numeric types.

}

```
@APIEntry{int
luaL_dofile (lua_State *L, const char *filename);|
@apii{0,?,m}
```

Loads and runs the given file.

It is defined as the following macro:

```
@verbatim{
(luaL_loadfile(L, filename) || lua_pcall(L, 0, LUA_MULTRET, 0))
}
It @N{returns 0} (@Lid{LUA_OK}) if there are no errors,
or 1 in case of errors.

}
```

```
@APIEntry{int luaL_dostring (lua_State *L, const char *str);|
@apii{0,?,-}
```

Loads and runs the given string.

It is defined as the following macro:

```
@verbatim{
(luaL_loadstring(L, str) || lua_pcall(L, 0, LUA_MULTRET, 0))
}
It @N{returns 0} (@Lid{LUA_OK}) if there are no errors,
or 1 in case of errors.

}
```

```
@APIEntry{int luaL_error (lua_State *L, const char *fmt, ...);|
@apii{0,0,v}
```

Raises an error.

The error message format is given by @id{fmt}

plus any extra arguments,

following the same rules of @Lid{lua\_pushfstring}.

It also adds at the beginning of the message the file name and

the line number where the error occurred,

if this information is available.

This function never returns,

but it is an

idiom to use it in @N{C functions}

as @T{return luaL\_error(@rep{args})}.

```
}
```

```
@APIEntry{int luaL_execresult (lua_State *L, int stat);|
@apii{0,3,m}
```

This function produces the return values for process-related functions in the standard library (@Lid{os.execute} and @Lid{io.close}).

}

```
@APIEntry{
int luaL_fileresult (lua_State *L, int stat, const char *fname);|
@apii{0,1|3,m}
```

This function produces the return values for file-related functions in the standard library (@Lid{io.open}, @Lid{os.rename}, @Lid{file:seek}, etc.).

}

```
@APIEntry{int luaL_getmetafield (lua_State *L, int obj, const char *e);|
@apii{0,0|1,m}
```

Pushes onto the stack the field @id{e} from the metatable of the object at index @id{obj} and returns the type of the pushed value. If the object does not have a metatable, or if the metatable does not have this field, pushes nothing and returns @id{LUA\_TNIL}.

}

```
@APIEntry{int luaL_getmetatable (lua_State *L, const char *tname);|
@apii{0,1,m}
```

Pushes onto the stack the metatable associated with the name @id{tname} in the registry @seeC{luaL\_newmetatable}, or @nil if there is no metatable associated with that name. Returns the type of the pushed value.

}

```
@APIEntry{int luaL_getsubtable (lua_State *L, int idx, const char *fname);|
@apii{0,1,e}
```

Ensures that the value @T{t[fname]}, where @id{t} is the value at index @id{idx}, is a table, and pushes that table onto the stack. Returns true if it finds a previous table there and false if it creates a new table.

```
}
```

```
@APIEntry{  
const char *luaL_gsub (lua_State *L,  
    const char *s,  
    const char *p,  
    const char *r);|  
@apii{0,1,m}
```

Creates a copy of string @id{s},  
replacing any occurrence of the string @id{p}  
with the string @id{r}.  
Pushes the resulting string on the stack and returns it.

```
}
```

```
@APIEntry{lua_Integer luaL_len (lua_State *L, int index);|  
@apii{0,0,e}
```

Returns the @Q{length} of the value at the given index  
as a number;  
it is equivalent to the  
@Char{#} operator in Lua @see{len-op}.  
Raises an error if the result of the operation is not an integer.  
(This case can only happen through metamethods.)

```
}
```

```
@APIEntry{  
int luaL_loadbuffer (lua_State *L,  
    const char *buff,  
    size_t sz,  
    const char *name);|  
@apii{0,1,-}
```

Equivalent to @Lid{luaL\_loadbufferx} with @id{mode} equal to @id{NULL}.

```
}
```

```
@APIEntry{  
int luaL_loadbufferx (lua_State *L,  
    const char *buff,  
    size_t sz,  
    const char *name,  
    const char *mode);|  
@apii{0,1,-}
```

Loads a buffer as a Lua chunk.

This function uses @Lid{lua\_load} to load the chunk in the buffer pointed to by @id{buff} with size @id{sz}.

This function returns the same results as @Lid{lua\_load}.

@id{name} is the chunk name,

used for debug information and error messages.

The string @id{mode} works as in the function @Lid{lua\_load}.

}

```
@APIEntry{int luaL_loadfile (lua_State
    *L, const char *filename);|
    @apii{0,1,m}
```

Equivalent to @Lid{luaL\_loadfilex} with @id{mode} equal to @id{NULL}.

}

```
@APIEntry{int luaL_loadfilex (lua_State *L, const char *filename,
    const char *mode);|
    @apii{0,1,m}
```

Loads a file as a Lua chunk.

This function uses @Lid{lua\_load} to load the chunk in the file named @id{filename}.

If @id{filename} is @id{NULL},

then it loads from the standard input.

The first line in the file is ignored if it starts with a @T{#}.

The string @id{mode} works as in the function @Lid{lua\_load}.

This function returns the same results as @Lid{lua\_load}

or @Lid{LUA\_ERRFILE} for file-related errors.

As @Lid{lua\_load}, this function only loads the chunk;

it does not run it.

}

```
@APIEntry{int luaL_loadstring (lua_State *L, const char *s);|
    @apii{0,1,-}
```

Loads a string as a Lua chunk.

This function uses @Lid{lua\_load} to load the chunk in the zero-terminated string @id{s}.

This function returns the same results as

@Lid{lua\_load}.

Also as @Lid{lua\_load}, this function only loads the chunk;  
it does not run it.

}

```
@APIEntry{void luaL_newlib (lua_State *L, const luaL_Reg l[]);|
@apii{0,1,m}}
```

Creates a new table and registers there  
the functions in the list @id{1}.

It is implemented as the following macro:

```
@verbatim{
(luaL_newlibtable(L,l), luaL_setfuncs(L,l,0))
}
```

The array @id{1} must be the actual array,  
not a pointer to it.

}

```
@APIEntry{void luaL_newlibtable (lua_State *L, const luaL_Reg l[]);|
@apii{0,1,m}}
```

Creates a new table with a size optimized  
to store all entries in the array @id{1}  
(but does not actually store them).

It is intended to be used in conjunction with @Lid{luaL\_setfuncs}  
@seeF{luaL\_newlib}.

It is implemented as a macro.

The array @id{1} must be the actual array,  
not a pointer to it.

}

```
@APIEntry{int luaL_newmetatable (lua_State *L, const char *tname);|
@apii{0,1,m}}
```

If the registry already has the key @id{tname},  
returns 0.

Otherwise,  
creates a new table to be used as a



metatable for userdata,  
adds to this new table the pair @T{\_\_name = tname},  
adds to the registry the pair @T{[tname] = new table},  
and returns 1.

In both cases,  
the function pushes onto the stack the final value associated  
with @id{tname} in the registry.

}

@APIEntry{lua\_State \*luaL\_newstate (void);|  
@apii{0,0,-}}

Creates a new Lua state.  
It calls @Lid{lua\_newstate} with an  
allocator based on the @N{ISO C} allocation functions  
and then sets a warning function and a panic function @see{C-error}  
that print messages to the standard error output.

Returns the new state,  
or @id{NULL} if there is a @x{memory allocation error}.

}

@APIEntry{void luaL\_openlibs (lua\_State \*L);|  
@apii{0,0,e}}

Opens all standard Lua libraries into the given state.

}

@APIEntry{  
T luaL\_opt (L, func, arg, dflt);|  
@apii{0,0,-}}

This macro is defined as follows:

@verbatim{  
(lua\_isnoneornil(L,(arg)) ? (dflt) : func(L,(arg)))  
}

In words, if the argument @id{arg} is nil or absent,  
the macro results in the default

@id{dflt}.

Otherwise, it results in the result of calling @id{func}  
with the state @id{L} and the argument index @id{arg} as  
arguments.

Note that it evaluates the expression @id{dflt} only if needed.

}

```
@APIEntry{
lua_Integer luaL_optinteger (lua_State *L,
                           int arg,
                           lua_Integer d);|
@apii{0,0,v}
```

If the function argument @id{arg} is an integer  
(or it is convertible to an integer),  
returns this integer.  
If this argument is absent or is @nil,  
returns @id{d}.  
Otherwise, raises an error.

}

```
@APIEntry{
const char *luaL_optlstring (lua_State *L,
                           int arg,
                           const char *d,
                           size_t *l);|
@apii{0,0,v}
```

If the function argument @id{arg} is a string,  
returns this string.  
If this argument is absent or is @nil,  
returns @id{d}.  
Otherwise, raises an error.

If @id{l} is not @id{NULL},  
fills its referent with the result's length.  
If the result is @id{NULL}  
(only  
possible when returning @id{d} and @T{d == NULL}),  
its length is considered zero.

This function uses @Lid{lua\_tolstring} to get its result,  
so all conversions and caveats of that function apply here.

}

```
@APIEntry{lua_Number luaL_optnumber (lua_State *L, int arg, lua_Number d);|
@apii{0,0,v}
```

If the function argument @id{arg} is a number,  
returns this number as a @id{lua\_Number}.  
If this argument is absent or is @nil,

returns @id{d}.

Otherwise, raises an error.

}

@APIEntry{  
const char \*luaL\_optstring (lua\_State \*L,  
int arg,  
const char \*d);|  
@apii{0,0,v}

If the function argument @id{arg} is a string,  
returns this string.

If this argument is absent or is @nil,  
returns @id{d}.

Otherwise, raises an error.

}

@APIEntry{char \*luaL\_prepbuffer (luaL\_Buffer \*B);|  
@apii{?,?,m}

Equivalent to @Lid{luaL\_prepbuffsize}  
with the predefined size @defid{LUAL\_BUFFERSIZE}.

}

@APIEntry{char \*luaL\_prepbuffsize (luaL\_Buffer \*B, size\_t sz);|  
@apii{?,?,m}

Returns  
an address to a space of size @id{sz}  
where you can copy a string to be added to buffer @id{B}  
@seeC{luaL\_Buffer}.

After copying the string into this space you must call  
@Lid{luaL\_addsize} with the size of the string to actually add  
it to the buffer.

}

@APIEntry{void luaL\_pushfail (lua\_State \*L);|  
@apii{0,1,-}

Pushes the @fail value onto the stack @see{libraries}.

}

@APIEntry{void luaL\_pushresult (luaL\_Buffer \*B);|

@apii{?,1,m}

Finishes the use of buffer @id{B} leaving the final string on the top of the stack.

}

@APIEntry{void luaL\_pushresultsz (luaL\_Buffer \*B, size\_t sz);|  
@apii{?,1,m}

Equivalent to the sequence @Lid{luaL\_addsize}, @Lid{luaL\_pushresult}.

}

@APIEntry{int luaL\_ref (lua\_State \*L, int t);|  
@apii{1,0,m}

Creates and returns a @def{reference},  
in the table at index @id{t},  
for the object on the top of the stack (and pops the object).

A reference is a unique integer key.  
As long as you do not manually add integer keys into the table @id{t},  
@Lid{luaL\_ref}  
ensures the uniqueness of the key it returns.  
You can retrieve an object referred by the reference @id{r}  
by calling @T{lua\_rawgeti(L, t, r)}.  
The function @Lid{luaL\_unref} frees a reference.

If the object on the top of the stack is @nil,  
@Lid{luaL\_ref} returns the constant @defid{LUA\_REFNIL}.  
The constant @defid{LUA\_NOREF} is guaranteed to be different  
from any reference returned by @Lid{luaL\_ref}.

}

@APIEntry{  
typedef struct luaL\_Reg {  
const char \*name;  
lua\_CFunction func;  
} luaL\_Reg;  
|

Type for arrays of functions to be registered by  
@Lid{luaL\_setfuncs}.  
@id{name} is the function name and @id{func} is a pointer to  
the function.  
Any array of @Lid{luaL\_Reg} must end with a sentinel entry

in which both @id{name} and @id{func} are @id{NULL}.

}

```
@APIEntry{  
void luaL_requiref (lua_State *L, const char *modname,  
                  lua_CFunction openf, int glb);|  
@apii{0,1,e}
```

If @T{package.loaded[modname]} is not true,  
calls the function @id{openf} with the string @id{modname}  
as an argument  
and sets the call result to @T{package.loaded[modname]},  
as if that function has been called through @Lid{require}.

If @id{glb} is true,  
also stores the module into the global @id{modname}.

Leaves a copy of the module on the stack.

}

```
@APIEntry{void luaL_setfuncs (lua_State *L, const luaL_Reg *l, int nup);|  
@apii{nup,0,m}
```

Registers all functions in the array @id{l}  
@seeC{luaL\_Reg} into the table on the top of the stack  
(below optional upvalues, see next).

When @id{nup} is not zero,  
all functions are created with @id{nup} upvalues,  
initialized with copies of the @id{nup} values  
previously pushed on the stack  
on top of the library table.  
These values are popped from the stack after the registration.

A function with a @id{NULL} value represents a placeholder,  
which is filled with @false.

}

```
@APIEntry{void luaL_setmetatable (lua_State *L, const char *tname);|  
@apii{0,0,-}
```

Sets the metatable of the object on the top of the stack  
as the metatable associated  
with name @id{tname}  
in the registry @seeC{luaL\_newmetatable}.

```
}
```

```
@APIEntry{  
typedef struct luaL_Stream {  
    FILE *f;  
    lua_CFunction closef;  
} luaL_Stream;  
|
```

The standard representation for @x{file handles}  
used by the standard I/O library.

A file handle is implemented as a full userdata,  
with a metatable called @id{LUA\_FILEHANDLE}  
(where @id{LUA\_FILEHANDLE} is a macro with the actual metatable's name).  
The metatable is created by the I/O library  
@seeF{luaL\_newmetatable}.

This userdata must start with the structure @id{luaL\_Stream};  
it can contain other data after this initial structure.  
The field @id{f} points to the corresponding C stream  
(or it can be @id{NULL} to indicate an incompletely created handle).  
The field @id{closef} points to a Lua function  
that will be called to close the stream  
when the handle is closed or collected;  
this function receives the file handle as its sole argument and  
must return either a true value, in case of success,  
or a false value plus an error message,  
in case of error.  
Once Lua calls this field,  
it changes the field value to @id{NULL}  
to signal that the handle is closed.

```
}
```

```
@APIEntry{void *luaL_testudata (lua_State *L, int arg, const char *tname);|  
@apii{0,0,m}
```

This function works like @Lid{luaL\_checkudata},  
except that, when the test fails,  
it returns @id{NULL} instead of raising an error.

```
}
```

```
@APIEntry{const char *luaL_tolstring (lua_State *L, int idx, size_t *len);|  
@apii{0,1,e}
```

Converts any Lua value at the given index to a @N{C string} in a reasonable format.

The resulting string is pushed onto the stack and also returned by the function @see{constchar}.

If @id{len} is not @id{NULL}, the function also sets @T{\*len} with the string length.

If the value has a metatable with a @idx{\_\_tostring} field, then @id{luaL\_tolstring} calls the corresponding metamethod with the value as argument, and uses the result of the call as its result.

}

```
@APIEntry{
void luaL_traceback (lua_State *L, lua_State *L1, const char *msg,
                    int level);|
@apii{0,1,m}
```

Creates and pushes a traceback of the stack @id{L1}.

If @id{msg} is not @id{NULL}, it is appended at the beginning of the traceback.

The @id{level} parameter tells at which level to start the traceback.

}

```
@APIEntry{int luaL_typeerror (lua_State *L, int arg, const char *tname);|
@apii{0,0,v}
```

Raises a type error for the argument @id{arg}

of the @N{C function} that called it, using a standard message;

@id{tname} is a @Q{name} for the expected type.

This function never returns.

}

```
@APIEntry{const char *luaL_typename (lua_State *L, int index);|
@apii{0,0,-}
```

Returns the name of the type of the value at the given index.

}

```
@APIEntry{void luaL_unref (lua_State *L, int t, int ref);|
@apii{0,0,-}
```

Releases the reference @id{ref} from the table at index @id{t}  
@seeC{luaL\_ref}.

The entry is removed from the table,  
so that the referred object can be collected.  
The reference @id{ref} is also freed to be used again.

If @id{ref} is @Lid{LUA\_NOREF} or @Lid{LUA\_REFNIL},  
@Lid{luaL\_unref}  
does nothing.

}

@APIEntry{void luaL\_where (lua\_State \*L, int lvl);|  
@apii{0,1,m}}

Pushes onto the stack a string identifying the current position  
of the control at level @id{lvl} in the call stack.

Typically this string has the following format:

@verbatim{  
@rep{chunkname}:@rep{currentline}:  
}

@N{Level 0} is the running function,  
@N{level 1} is the function that called the running function,  
etc.

This function is used to build a prefix for error messages.

}

}

}

@C{-----}  
@sect1{libraries| @title{The Standard Libraries}}

@simplesect{

The standard Lua libraries provide useful functions  
that are implemented @N{in C} through the @N{C API}.  
Some of these functions provide essential services to the language  
(e.g., @Lid{type} and @Lid{getmetatable});  
others provide access to outside services (e.g., I/O);  
and others could be implemented in Lua itself,  
but that for different reasons  
deserve



an implementation in C (e.g., `@Lid{table.sort}`).

All libraries are implemented through the official `@N{C API}` and are provided as separate `@N{C modules}`.

Unless otherwise noted,

these library functions do not adjust its number of arguments to its expected parameters.

For instance, a function documented as `@T{foo(arg)}` should not be called without an argument.

The notation `@fail` means a false value representing some kind of failure.

(Currently, `@fail` is equal to `@nil`,

but that may change in future versions.

The recommendation is to always test the success of these functions with `@T{(not status)}`, instead of `@T{(status == nil)}`.)

Currently, Lua has the following standard libraries:

`@itemize{`

`@item{@link{predefined|basic library};}`

`@item{@link{corolib|coroutine library};}`

`@item{@link{packlib|package library};}`

`@item{@link{strlib|string manipulation};}`

`@item{@link{utf8|basic UTF-8 support};}`

`@item{@link{tablib|table manipulation};}`

`@item{@link{mathlib|mathematical functions}  
(sin, log, etc.);}`

`@item{@link{iolib|input and output};}`

`@item{@link{oslib|operating system facilities};}`

`@item{@link{debuglib|debug facilities};.}`

`}`

Except for the basic and the package libraries, each library provides all its functions as fields of a global table or as methods of its objects.

To have access to these libraries,

the @N{C host} program should call the @Lid{luaL\_openlibs} function, which opens all standard libraries.

Alternatively,

the host program can open them individually by using

@Lid{luaL\_requiref} to call

@defid{luaopen\_base} (for the basic library),

@defid{luaopen\_package} (for the package library),

@defid{luaopen\_coroutine} (for the coroutine library),

@defid{luaopen\_string} (for the string library),

@defid{luaopen\_utf8} (for the UTF-8 library),

@defid{luaopen\_table} (for the table library),

@defid{luaopen\_math} (for the mathematical library),

@defid{luaopen\_io} (for the I/O library),

@defid{luaopen\_os} (for the operating system library),

and @defid{luaopen\_debug}

(for the debug library).

These functions are declared in @defid{lualib.h}.

}

## @sect2{predefined| @title{Basic Functions}}

The basic library provides core functions to Lua.

If you do not include this library in your application, you should check carefully whether you need to provide implementations for some of its facilities.

@LibEntry{assert (v [, message])|

Raises an error if

the value of its argument @id{v} is false (i.e., @nil or @false); otherwise, returns all its arguments.

In case of error,

@id{message} is the error object;

when absent, it defaults to @St{assertion failed!}

}

@LibEntry{collectgarbage ([opt [, arg]])|

This function is a generic interface to the garbage collector.

It performs different functions according to its first argument, @id{opt}:

@description{

@item{@St{collect}|

Performs a full garbage-collection cycle.

This is the default option.

}

@item{@St{stop}|

Stops automatic execution of the garbage collector.

The collector will run only when explicitly

invoked,

until a call to restart it.

}

@item{@St{restart}|

Restarts automatic execution of the garbage collector.

}

@item{@St{count}|

Returns the total memory in use by Lua in Kbytes.

The value has a fractional part,

so that it multiplied by 1024

gives the exact number of bytes in use by Lua.

}

@item{@St{step}|

Performs a garbage-collection step.

The step @Q{size} is controlled by @id{arg}.

With a zero value,

the collector will perform one basic (indivisible) step.

For non-zero values,

the collector will perform as if that amount of memory

(in Kbytes) had been allocated by Lua.

Returns @true if the step finished a collection cycle.

}

@item{@St{isrunning}|

Returns a boolean that tells whether the collector is running

(i.e., not stopped).

}

@item{@St{incremental}|

Change the collector mode to incremental.

This option can be followed by three numbers:

the garbage-collector pause,

the step multiplier,

and the step size @see{incmode}.

A zero means to not change that value.

}

@item{@St{generational}|

Change

the collector mode to generational.

This option can be followed by two numbers:

the garbage-collector minor multiplier

and the major multiplier @see{genmode}.

A zero means to not change that value.

}

}

See @See{GC} for more details about garbage collection

and some of these options.

This function should not be called by a finalizer.

}

@LibEntry{dofile ([filename])|

Opens the named file and executes its content as a Lua chunk.

When called without arguments,

@id{dofile} executes the content of the standard input (@id{stdin}).

Returns all values returned by the chunk.

In case of errors, @id{dofile} propagates the error

to its caller.

(That is, @id{dofile} does not run in protected mode.)

}

@LibEntry{error (message [, level])|

Raises an error @see{error} with @id{message} as the error object.

This function never returns.

Usually, @id{error} adds some information about the error position at the beginning of the message, if the message is a string.

The @id{level}

argument specifies how to get the error position.

With @N{level 1} (the default), the error position is where the

@id{error} function was called.

@N{Level 2} points the error to where the function

that called @id{error} was called; and so on.

Passing a @N{level 0} avoids the addition of error position information to the message.

}

@LibEntry{\_G|

A global variable (not a function) that

holds the @x{global environment} @see{globalenv}.

Lua itself does not use this variable;

changing its value does not affect any environment,

nor vice versa.

}

@LibEntry{getmetatable (object)|

If @id{object} does not have a metatable, returns @nil.

Otherwise,

if the object's metatable has a @idx{\_\_metatable} field,  
returns the associated value.

Otherwise, returns the metatable of the given object.

}

@LibEntry{ipairs (t)|

Returns three values (an iterator function, the table @id{t}, and 0)  
so that the construction

@verbatim{

for i,v in ipairs(t) do @rep{body} end

}

will iterate over the key@En{ }value

pairs

(@T{1,t[1]}), (@T{2,t[2]}), @ldots,

up to the first absent index.

}

@LibEntry{load (chunk [, chunkname [, mode [, env]])|

Loads a chunk.

If @id{chunk} is a string, the chunk is this string.

If @id{chunk} is a function,

@id{load} calls it repeatedly to get the chunk pieces.

Each call to @id{chunk} must return a string that concatenates  
with previous results.

A return of an empty string, @nil, or no value signals the end of the chunk.

If there are no syntactic errors,

@id{load} returns the compiled chunk as a function;

otherwise, it returns @fail plus the error message.

When you load a main chunk,

the resulting function will always have exactly one upvalue,

the @id{\_ENV} variable @see{globalenv}.

However,

when you load a binary chunk created from a function @seeF{string.dump},

the resulting function can have an arbitrary number of upvalues, and there is no guarantee that its first upvalue will be the `@id{_ENV}` variable.  
(A non-main function may not even have an `@id{_ENV}` upvalue.)

Regardless, if the resulting function has any upvalues, its first upvalue is set to the value of `@id{env}`, if that parameter is given, or to the value of the `@x{global environment}`. Other upvalues are initialized with `@nil`. All upvalues are fresh, that is, they are not shared with any other function.

`@id{chunkname}` is used as the name of the chunk for error messages and debug information `@see{debugI}`.  
When absent, it defaults to `@id{chunk}`, if `@id{chunk}` is a string, or to `@St{=(load)}` otherwise.

The string `@id{mode}` controls whether the chunk can be text or binary (that is, a precompiled chunk).  
It may be the string `@St{b}` (only `@x{binary chunk}s`), `@St{t}` (only text chunks), or `@St{bt}` (both binary and text).  
The default is `@St{bt}`.

It is safe to load malformed binary chunks;  
`@id{load}` signals an appropriate error.  
However,  
Lua does not check the consistency of the code inside binary chunks; running maliciously crafted bytecode can crash the interpreter.

}

`@LibEntry{loadfile`  
`([filename [, mode [, env]]])`

Similar to `@Lid{load}`,  
but gets the chunk from file `@id{filename}`  
or from the standard input,  
if no file name is given.

}

`@LibEntry{next (table [, index])`

Allows a program to traverse all fields of a table.

Its first argument is a table and its second argument is an index in this table.  
 A call to `@id{next}` returns the next index of the table and its associated value.  
 When called with `@nil` as its second argument, `@id{next}` returns an initial index and its associated value.  
 When called with the last index, or with `@nil` in an empty table, `@id{next}` returns `@nil`.  
 If the second argument is absent, then it is interpreted as `@nil`.  
 In particular, you can use `@T{next(t)}` to check whether a table is empty.

The order in which the indices are enumerated is not specified, `@emph{even for numeric indices}`.  
 (To traverse a table in numerical order, use a numerical `@Rw{for}`.)

You should not assign any value to a non-existent field in a table during its traversal.  
 You may however modify existing fields.  
 In particular, you may set existing fields to `nil`.

}

`@LibEntry{pairs (t)|`

If `@id{t}` has a metamethod `@idx{__pairs}`, calls it with `@id{t}` as argument and returns the first three results from the call.

Otherwise,  
 returns three values: the `@Lid{next}` function, the table `@id{t}`, and `@nil`, so that the construction  
`@verbatim{  
 for k,v in pairs(t) do @rep{body} end  
}`  
 will iterate over all `key@En{}` value pairs of table `@id{t}`.

See function `@Lid{next}` for the caveats of modifying the table during its traversal.

}

`@LibEntry{pcall (f [, arg1, @Cdots])|`

Calls the function `@id{f}` with  
 the given arguments in `@emphx{protected mode}`.  
 This means that any error `@N{inside @T{f}}` is not propagated;  
 instead, `@id{pcall}` catches the error  
 and returns a status code.  
 Its first result is the status code (a boolean),  
 which is `@true` if the call succeeds without errors.  
 In such case, `@id{pcall}` also returns all results from the call,  
 after this first result.  
 In  
 case of any error, `@id{pcall}` returns `@false` plus the error object.  
 Note that errors caught by `@id{pcall}` do not call a message handler.

}

`@LibEntry{print (@Cdots)|`  
 Receives any number of arguments  
 and prints their values to `@id{stdout}`,  
 converting each argument to a string  
 following the same rules of `@Lid{tostring}`.

The function `@id{print}` is not intended for formatted output,  
 but only as a quick way to show a value,  
 for instance for debugging.  
 For complete control over the output,  
 use `@Lid{string.format}` and `@Lid{io.write}`.

}

`@LibEntry{rawequal (v1, v2)|`  
 Checks whether `@id{v1}` is equal to `@id{v2}`,  
 without invoking the `@idx{__eq}` metamethod.  
 Returns a boolean.

}

`@LibEntry{rawget (table, index)|`  
 Gets the real value of `@T{table[index]}`,  
 without using the `@idx{__index}` metavalue.  
`@id{table}` must be a table;  
`@id{index}` may be any value.

}

`@LibEntry{rawlen (v)|`  
 Returns the length of the object `@id{v}`,  
 which must be a table or a string,  
 without invoking the `@idx{__len}` metamethod.



Returns  
an integer.

}

@LibEntry{rawset (table, index, value)|  
Sets the real value of @T{table[index]} to @id{value},  
without using the @idx{\_\_newindex} metavalue.  
@id{table} must be a table,  
@id{index} any value different from @nil and @x{NaN},  
and @id{value} any Lua value.

This function returns @id{table}.

}

@LibEntry{select (index, @Cdots)|

If @id{index} is a number,  
returns all arguments after argument number @id{index};  
a negative number indexes from the end (@num{-1} is the last argument).  
Otherwise, @id{index} must be the string @T{"#"},  
and @id{select} returns the total number of extra arguments it received.

}

@LibEntry{setmetatable (table, metatable)|

Sets the metatable for the given table.  
If @id{metatable} is @nil,  
removes the metatable of the given table.  
If the original metatable has a @idx{\_\_metatable} field,  
raises an error.

This function returns @id{table}.

To change the metatable of other types from Lua code,  
you must use the @link{debuglib|debug library}.

}

@LibEntry{tonumber  
(e [, base])|

When called with no @id{base},  
@id{tonumber} tries to convert its argument to a number.  
If the argument is already a number or  
a string convertible to a number,

then `@id{tonumber}` returns this number;  
otherwise, it returns `@fail`.

The conversion of strings can result in integers or floats,  
according to the lexical conventions of Lua `@see{lexical}`.  
The string may have leading and trailing spaces and a sign.

When called with `@id{base}`,  
then `@id{e}` must be a string to be interpreted as  
an integer numeral in that base.  
The base may be any integer between 2 and 36, inclusive.  
In bases `@N{above 10}`, the letter `@Char{A}` (in either upper or lower case)  
`@N{represents 10}`, `@Char{B}` `@N{represents 11}`, and so forth,  
with `@Char{Z}` representing 35.  
If the string `@id{e}` is not a valid numeral in the given base,  
the function returns `@fail`.

}

`@LibEntry{tostring (v)|`

Receives a value of any type and  
converts it to a string in a human-readable format.

If the metatable  
of `@id{v}` has a `@idx{__tostring}` field,  
then `@id{tostring}` calls the corresponding value  
with `@id{v}` as argument,  
and uses the result of the call as its result.  
Otherwise, if the metatable of `@id{v}` has a `@idx{__name}` field  
with a string value,  
`@id{tostring}` may use that string in its final result.

For complete control of how numbers are converted,  
use `@Lid{string.format}`.

}

`@LibEntry{type (v)|`

Returns the type of its only argument, coded as a string.  
The possible results of this function are  
`@St{nil}` (a string, not the value `@nil`),  
`@St{number}`,  
`@St{string}`,  
`@St{boolean}`,  
`@St{table}`,  
`@St{function}`,

```
@St{thread},  
and @St{userdata}.
```

```
}
```

```
@LibEntry{_VERSION|
```

A global variable (not a function) that holds a string containing the running Lua version. The current value of this variable is @St{Lua 5.4}.

```
}
```

```
@LibEntry{warn (msg1, @Cdots)|
```

Emits a warning with a message composed by the concatenation of all its arguments (which should be strings).

By convention,  
a one-piece message starting  
with @Char{@At}  
is intended to be a @emph{control message},  
which is a message to the warning system itself.  
In particular, the standard warning function in Lua  
recognizes the control messages @St{@At{}off},  
to stop the emission of warnings,  
and @St{@At{}on}, to (re)start the emission;  
it ignores unknown control messages.

```
}
```

```
@LibEntry{xpcall (f, msgh [, arg1, @Cdots])|
```

This function is similar to @Lid{pcall},  
except that it sets a new @x{message handler} @id{msgh}.

```
}
```

```
}
```

```
@sect2{corolib| @title{Coroutine Manipulation}
```

This library comprises the operations to manipulate coroutines,  
which come inside the table @defid{coroutine}.  
See @See{coroutine} for a general description of coroutines.

```
@LibEntry{coroutine.close (co)|
```

Closes coroutine `@id{co}`,  
that is,  
closes all its pending to-be-closed variables  
and puts the coroutine in a dead state.  
The given coroutine must be dead or suspended.  
In case of error  
(either the original error that stopped the coroutine or  
errors in closing  
methods),  
returns `@false` plus the error object;  
otherwise returns `@true`.

}

`@LibEntry{coroutine.create (f)}`

Creates a new coroutine, with body `@id{f}`.  
`@id{f}` must be a function.  
Returns this new coroutine,  
an object with type `@T{"thread"}`.

}

`@LibEntry{coroutine.isyieldable ([co])}`

Returns `@true` when the coroutine `@id{co}` can yield.  
The default for `@id{co}` is the running coroutine.

A coroutine is yieldable if it is not the main thread and  
it is not inside a non-yieldable `@N{C function}`.

}

`@LibEntry{coroutine.resume (co [, val1, @Cdots])}`

Starts or continues the execution of coroutine `@id{co}`.  
The first time you resume a coroutine,  
it starts running its body.  
The values `@id{val1}`, `@ldots` are passed  
as the arguments to the body function.  
If the coroutine has yielded,  
`@id{resume}` restarts it;  
the values `@id{val1}`, `@ldots` are passed  
as the results from the yield.

If the coroutine runs without any errors,  
`@id{resume}` returns `@true` plus any values passed to `@id{yield}`

(when the coroutine  
yields) or any values returned by the body function  
(when the coroutine terminates).  
If there is any error,  
@id{resume} returns @false plus the error message.

}

@LibEntry{coroutine.running ()|

Returns the running coroutine plus a boolean,  
@true when the running coroutine is the main one.

}

@LibEntry{coroutine.status (co)|

Returns the status of the coroutine @id{co}, as a string:  
@T{"running"},  
if the coroutine is running  
(that is, it is the one that called @id{status});  
@T{"suspended"}, if the coroutine is suspended in a call to @id{yield},  
or if it has not started running yet;  
@T{"normal"} if the coroutine is active but not running  
(that is, it has resumed another coroutine);  
and @T{"dead"} if the coroutine has finished its body function,  
or if it has stopped with an error.

}

@LibEntry{coroutine.wrap (f)|

Creates a new coroutine, with body @id{f};  
@id{f} must be a function.  
Returns a function that resumes the coroutine each time it is called.  
Any arguments passed to this  
function behave as the  
extra arguments to @id{resume}.  
The function returns the same values returned by @id{resume},  
except the first boolean.  
In case of error,  
the function closes the coroutine and propagates the error.

}

@LibEntry{coroutine.yield (@Cdots)|

Suspends the execution of the calling coroutine.

Any arguments to `@id{yield}` are passed as extra results to `@id{resume}`.

}

}

`@sect2{packlib| @title{Modules}}`

The package library provides basic facilities for loading modules in Lua.

It exports one function directly in the global environment:

`@Lid{require}`.

Everything else is exported in the table `@defid{package}`.

`@LibEntry{require (modname)|`

Loads the given module.

The function starts by looking into the `@Lid{package.loaded}` table to determine whether `@id{modname}` is already loaded.

If it is, then `@id{require}` returns the value stored at `@T{package.loaded[modname]}`.

(The absence of a second result in this case signals that this call did not have to load the module.)

Otherwise,

it tries to find a `@emph{loader}` for the module.

To find a loader,

`@id{require}` is guided by the table `@Lid{package.searchers}`.

Each item in this table is a search function, that searches for the module in a particular way.

By changing this table,

we can change how `@id{require}` looks for a module.

The following explanation is based on the default configuration for `@Lid{package.searchers}`.

First `@id{require}` queries `@T{package.preload[modname]}`.

If it has a value,

this value (which must be a function) is the loader.

Otherwise `@id{require}` searches for a Lua loader using the path stored in `@Lid{package.path}`.

If that also fails, it searches for a `@N{C loader}` using the path stored in `@Lid{package.cpath}`.

If that also fails,

it tries an `@emph{all-in-one}` loader `@seeF{package.searchers}`.

Once a loader is found,

`@id{require}` calls the loader with two arguments:

`@id{modname}` and an extra value,  
a `@emph{loader data}`,  
also returned by the searcher.  
The loader data can be any value useful  
to the module;  
for the default searchers,  
it indicates where the loader was found.  
(For instance, if the loader came from a file,  
this extra value is the file path.)  
If the loader returns any non-nil value,  
`@id{require}` assigns the returned value to `@T{package.loaded[modname]}`.  
If the loader does not return a non-nil value and  
has not assigned any value to `@T{package.loaded[modname]}`,  
then `@id{require}` assigns `@true` to this entry.  
In any case, `@id{require}` returns the  
final value of `@T{package.loaded[modname]}`.  
Besides that value, `@id{require}` also returns as a second result  
the loader data returned by the searcher,  
which indicates how `@id{require}` found the module.

If there is any error loading or running the module,  
or if it cannot find any loader for the module,  
then `@id{require}` raises an error.

}

`@LibEntry{package.config|`

A string describing some compile-time configurations for packages.

This string is a sequence of lines:

`@itemize{`

`@item{The first line is the @x{directory  
separator} string.`

Default is `@Char{\}` for `@x{Windows}` and `@Char{/}` for all other systems.}

`@item{The second line is the character that separates templates in a path.`

Default is `@Char{;}.}`

`@item{The third line is the string that marks the  
substitution points in a template.`

Default is `@Char{?}.}`

`@item{The fourth line is a string that, in a path in @x{Windows},  
is replaced by the executable's directory.`

Default is `@Char{!}.}`

`@item{The fifth line is a mark to ignore all text after it`

when building the `@id{luaopen_}` function name.

Default is `@Char{-}.}`

}

}

`@LibEntry{package.cpath|`

A string with the path used by `@Lid{require}`

to search for a `@N{C loader}`.

Lua initializes the `@N{C path}` `@Lid{package.cpath}` in the same way

it initializes the Lua path `@Lid{package.path}`,

using the environment variable `@defid{LUA_CPATH_5_4}`,

or the environment variable `@defid{LUA_CPATH}`,

or a default path defined in `@id{luaconf.h}`.

}

`@LibEntry{package.loaded|`

A table used by `@Lid{require}` to control

which

modules are already loaded.

When you require a module `@id{modname}` and

`@T{package.loaded[modname]}` is not false,

`@Lid{require}` simply returns the value stored there.

This variable is only a reference to the real table;

assignments to this variable do not change the

table used by `@Lid{require}`.

The real table is stored in the C registry `@see{registry}`,

indexed by the key `@defid{LUA_LOADED_TABLE}`, a string.

}

`@LibEntry{package.loadlib (libname, funcname)|`

Dynamically links the host program with the `@N{C library}` `@id{libname}`.

If `@id{funcname}` is `@St{*}`,

then it only links with the library,

making the symbols exported by the library

available to other dynamically linked libraries.

Otherwise,

it looks for a function `@id{funcname}` inside the library

and returns this function as a `@N{C function}`.



So, @id{funcname} must follow the @Lid{lua\_CFunction} prototype  
@seeC{lua\_CFunction}.

This is a low-level function.

It completely bypasses the package and module system.

Unlike @Lid{require},

it

does not perform any path searching and

does not automatically add extensions.

@id{libname} must be the complete file name of the @N{C library},

including if necessary a path and an extension.

@id{funcname} must be the exact name exported by the @N{C library}

(which may depend on the @N{C compiler} and linker used).

This functionality is not supported by @N{ISO C}.

As such, it is only available on some platforms

(Windows, Linux, Mac OS X, Solaris, BSD,

plus other Unix systems that support the @id{dlfcn} standard).

This function is inherently insecure,

as it allows Lua to call any function in any readable dynamic  
library in the system.

(Lua calls any function assuming the function

has a proper prototype and respects a proper protocol

@see{lua\_CFunction}.

Therefore,

calling an arbitrary function in an arbitrary dynamic library

more often than not results in an access violation.)

}

@LibEntry{package.path|

A string with the path used by @Lid{require}

to search for a Lua

loader.

At start-up, Lua initializes this variable with

the value of the environment variable @defid{LUA\_PATH\_5\_4} or

the environment variable @defid{LUA\_PATH} or

with a default path defined in @id{luaconf.h},

if those environment variables are not defined.

A @St{;;} in the value of the environment variable

is replaced by the default path.

}

@LibEntry{package.preload|

A table to store loaders for specific modules

@seeF{require}.

This variable is only a reference to the real table;

assignments to this variable do not change the

table used by @Lid{require}.

The real table is stored in the C registry @see{registry},

indexed by the key @defid{LUA\_PRELOAD\_TABLE}, a string.

}

@LibEntry{package.searchers|

A table used by @Lid{require} to control how to find modules.

Each entry in this table is a @def{searcher function}.

When looking for a module,

@Lid{require} calls each of these searchers in ascending order,  
with the module name (the argument given to @Lid{require}) as its  
sole argument.

If

the searcher finds the module,

it returns another function, the module @def{loader},

plus an extra value, a @emph{loader data},

that will be passed to that loader and

returned as a second result by @Lid{require}.

If it cannot find the module,

it returns a string explaining why

(or @nil if it has nothing to say).

Lua initializes this table with four searcher functions.

The first searcher simply looks for a loader in the

@Lid{package.preload} table.

The second searcher looks for a loader as a Lua library,

using the path stored at @Lid{package.path}.

The search is done as described in function @Lid{package.searchpath}.

The third searcher looks for a loader as a @N{C library},

using the path given by the variable @Lid{package.cpath}.

Again,

the search is done as described in function @Lid{package.searchpath}.

For instance,

if the @N{C path} is the string

@verbatim{

"/?.so;/??.dll;/usr/local/?/init.so"

}  
 the searcher for module @id{foo}  
 will try to open the files @T{./foo.so},  
 @T{./foo.dll},  
 and @T{/usr/local/foo/init.so}, in that order.  
 Once it finds a @N{C library},  
 this searcher first uses a dynamic link facility to link the  
 application with the library.  
 Then it tries to find a @N{C function} inside the library to  
 be used as the loader.  
 The name of this @N{C function} is the string @St{luaopen\_  
 concatenated with a copy of the module name where each dot  
 is replaced by an underscore.  
 Moreover, if the module name has a hyphen,  
 its suffix after (and including) the first hyphen is removed.  
 For instance, if the module name is @id{a.b.c-v2.1},  
 the function name will be @id{luaopen\_a\_b\_c}.

The fourth searcher tries an @def{all-in-one loader}.  
 It searches the @N{C path} for a library for  
 the root name of the given module.  
 For instance, when requiring @id{a.b.c},  
 it will search for a @N{C library} for @id{a}.  
 If found, it looks into it for an open function for  
 the submodule;  
 in our example, that would be @id{luaopen\_a\_b\_c}.  
 With this facility, a package can pack  
 several @N{C submodules}  
 into one single library,  
 with each submodule keeping its original open function.

All searchers except the first one (preload) return as the extra value  
 the file path where the module was found,  
 as returned by @Lid{package.searchpath}.  
 The first searcher always returns the string @St{:preload:}.

Searchers should raise no errors and have no side effects in Lua.  
 (They may have side effects in C,  
 for instance by linking the application with a library.)

}

@LibEntry{package.searchpath (name, path [, sep [, rep]])|

Searches for the given @id{name} in the given @id{path}.

A path is a string containing a sequence of  
 @emph{templates} separated by semicolons.

For each template,  
the function replaces each interrogation mark (if any)  
in the template with a copy of @id{name}  
wherein all occurrences of @id{sep}  
(a dot, by default)  
were replaced by @id{rep}  
(the system's directory separator, by default),  
and then tries to open the resulting file name.

For instance, if  
the path is the string  
@verbatim{  
"./?.lua;./?.lc;/usr/local/?/init.lua"  
}  
the search for the name @id{foo.a}  
will try to open the files  
@T{./foo/a.lua}, @T{./foo/a.lc}, and  
@T{/usr/local/foo/a/init.lua}, in that order.

Returns the resulting name of the first file that it can  
open in read mode (after closing the file),  
or @fail plus an error message if none succeeds.  
(This error message lists all file names it tried to open.)

}

}

@sect2{strlib| @title{String Manipulation}

@simplesect{

This library provides generic functions for string manipulation,  
such as finding and extracting substrings, and pattern matching.  
When indexing a string in Lua, the first character is at @N{position 1}  
(not @N{at 0}, as in C).  
Indices are allowed to be negative and are interpreted as indexing backwards,  
from the end of the string.  
Thus, the last character is at position @num{-1}, and so on.

The string library provides all its functions inside the table  
@defid{string}.  
It also sets a @x{metatable for  
strings}  
where the @idx{\_\_index} field points to the @id{string} table.  
Therefore, you can use the string functions in object-oriented style.  
For instance, @T{string.byte(s,i)}  
can be written as @T{s:byte(i)}.

The string library assumes one-byte character encodings.

`@LibEntry{string.byte (s [, i [, j]])}`

Returns the internal numeric codes of the characters `@T{s[i]}`,  
`@T{s[i+1]}`, `@ldots`, `@T{s[j]}`.

The default value for `@id{i}` `@N{is 1}`;

the default value for `@id{j}` `@N{is @id{i}}`.

These indices are corrected

following the same rules of function `@Lid{string.sub}`.

Numeric codes are not necessarily portable across platforms.

}

`@LibEntry{string.char (@Cdots)}`

Receives zero or more integers.

Returns a string with length equal to the number of arguments,  
in which each character has the internal numeric code equal  
to its corresponding argument.

Numeric codes are not necessarily portable across platforms.

}

`@LibEntry{string.dump (function [, strip])}`

Returns a string containing a binary  
representation

(a `@emph{binary chunk}`)

of the given function,

so that a later `@Lid{load}` on this string returns

a copy of the function (but with new upvalues).

If `@id{strip}` is a true value,

the binary representation may not include all debug information  
about the function,

to save space.

Functions with upvalues have only their number of upvalues saved.

When (re)loaded,

those upvalues receive fresh instances.

(See the `@Lid{load}` function for details about

how these upvalues are initialized.

You can use the debug library to serialize

and reload the upvalues of a function

in a way adequate to your needs.)

}

@LibEntry{string.find (s, pattern [, init [, plain]])|

Looks for the first match of

@id{pattern} @see{pm} in the string @id{s}.

If it finds a match, then @id{find} returns the indices @N{of @T{s}} where this occurrence starts and ends;

otherwise, it returns @fail.

A third, optional numeric argument @id{init} specifies

where to start the search;

its default value @N{is 1} and can

be negative.

A @true as a fourth, optional argument @id{plain}

turns off the pattern matching facilities,

so the function does a plain @Q{find substring} operation,

with no characters in @id{pattern} being considered magic.

If the pattern has captures,

then in a successful match

the captured values are also returned,

after the two indices.

}

@LibEntry{string.format (formatstring, @Cdots)|

Returns a formatted version of its variable number of arguments

following the description given in its first argument,

which must be a string.

The format string follows the same rules as the @ANSI{sprintf}.

The only differences are that the conversion specifiers and modifiers

@id{F}, @id{n}, @T{\*}, @id{h}, @id{L}, and @id{l} are not supported and that there is an extra specifier, @id{q}.

Both width and precision, when present,

are limited to two digits.

The specifier @id{q} formats booleans, nil, numbers, and strings

in a way that the result is a valid constant in Lua source code.

Booleans and nil

are written in the obvious way

(@id{true}, @id{false}, @id{nil}).

Floats are written in hexadecimal,

to preserve full precision.

A string is written between double quotes,

using escape sequences when necessary to ensure that

it can safely be read back by the Lua interpreter.

For instance, the call

```
@verbatim{
string.format('%q', 'a string with "quotes" and \n new line')
}
```

may produce the string:

```
@verbatim{
"a string with \"quotes\" and \
new line"
}
```

This specifier does not support modifiers (flags, width, precision).

The conversion specifiers

`@id{A}`, `@id{a}`, `@id{E}`, `@id{e}`, `@id{f}`,  
`@id{G}`, and `@id{g}` all expect a number as argument.

The specifiers `@id{c}`, `@id{d}`,

`@id{i}`, `@id{o}`, `@id{u}`, `@id{X}`, and `@id{x}`

expect an integer.

When Lua is compiled with a C89 compiler,  
the specifiers `@id{A}` and `@id{a}` (hexadecimal floats)  
do not support modifiers.

The specifier `@id{s}` expects a string;

if its argument is not a string,

it is converted to one following the same rules of `@Lid{tostring}`.

If

the specifier has any modifier,

the corresponding string argument should not contain `@x{embedded zeros}`.

The specifier `@id{p}` formats the pointer

returned by `@Lid{lua_topointer}`.

That gives a unique string identifier for tables, userdata,  
threads, strings, and functions.

For other values (numbers, nil, booleans),  
this specifier results in a string representing  
the pointer `@id{NULL}`.

}

```
@LibEntry{string.gmatch (s, pattern [, init])|
```

Returns an iterator function that,

each time it is called,

returns the next captures from `@id{pattern}` `@see{pm}`  
over the string `@id{s}`.

If `@id{pattern}` specifies no captures,

then the whole match is produced in each call.

A third, optional numeric argument `@id{init}` specifies  
where to start the search;

its default value `@N{is 1}` and can be negative.

As an example, the following loop  
will iterate over all the words from string `@id{s}`,  
printing one per line:

```
@verbatim{
s = "hello world from Lua"
for w in string.gmatch(s, "%a+") do
  print(w)
end
}
```

The next  
example collects all pairs `@T{key=value}` from the  
given string into a table:

```
@verbatim{
t = {}
s = "from=world, to=Lua"
for k, v in string.gmatch(s, "(%w+)=(%w+)") do
  t[k] = v
end
}
```

For this function, a caret `@Char{^}` at the start of a pattern does not  
work as an anchor, as this would prevent the iteration.

}

`@LibEntry{string.gsub (s, pattern, repl [, n])}`

Returns a copy of `@id{s}`

in which all (or the first `@id{n}`, if given)

occurrences of the `@id{pattern}` `@see{pm}` have been  
replaced by a replacement string specified by `@id{repl}`,  
which can be a string, a table, or a function.

`@id{gsub}` also returns, as its second value,  
the total number of matches that occurred.

The name `@id{gsub}` comes from `@emph{Global SUBstitution}`.

If `@id{repl}` is a string, then its value is used for replacement.

The `@N{character @T{}}` works as an escape character:

any sequence in `@id{repl}` of the form `@T{%%@rep{d}}`,

with `@rep{d}` between 1 and 9,

stands for the value of the `@rep{d}`-th captured substring;  
the

sequence `@T{%0}` stands for the whole match;

the sequence `@T{%%}` stands for a `@N{single @T{}}`.

If `@id{repl}` is a table, then the table is queried for every match,  
using the first capture as the key.

If `@id{repl}` is a function, then this function is called every time a



match occurs, with all captured substrings passed as arguments, in order.

In any case,  
if the pattern specifies no captures,  
then it behaves as if the whole pattern was inside a capture.

If the value returned by the table query or by the function call  
is a string or a number,  
then it is used as the replacement string;  
otherwise, if it is `@Rw{false}` or `@nil`,  
then there is no replacement  
(that is, the original match is kept in the string).

Here are some examples:

```
@verbatim{
```

```
x = string.gsub("hello world", "(%w+)", "%1 %1")
--> x="hello hello world world"
```

```
x = string.gsub("hello world", "%w+", "%0 %0", 1)
--> x="hello hello world"
```

```
x = string.gsub("hello world from Lua", "(%w+)%s*(%w+)", "%2 %1")
--> x="world hello Lua
from"
```

```
x = string.gsub("home = $HOME, user = $USER", "%$(%w+)", os.getenv)
--> x="home = /home/roberto, user = roberto"
```

```
x = string.gsub("4+5 = $return 4+5$", "%$(.)%$", function (s)
    return load(s)()
end)
--> x="4+5 = 9"
```

```
local t = {name="lua", version="5.4"}
x = string.gsub("$name-$version.tar.gz", "%$(%w+)", t)
--> x="lua-5.4.tar.gz"
}

}
```

```
@LibEntry{string.len (s)|
```

Receives a string and returns its length.  
The empty string `@T{""}` has length 0.  
Embedded zeros are counted,  
so `@T{"a\000bc\000"}` has length 5.

}

@LibEntry{string.lower (s)|

Receives a string and returns a copy of this string with all uppercase letters changed to lowercase.

All other characters are left unchanged.

The definition of what an uppercase letter is depends on the current locale.

}

@LibEntry{string.match (s, pattern [, init])|

Looks for the first @emph{match} of the @id{pattern} @see{pm} in the string @id{s}.

If it finds one, then @id{match} returns the captures from the pattern;

otherwise it

returns @fail.

If @id{pattern} specifies no captures, then the whole match is returned.

A third, optional numeric argument @id{init} specifies where to start the search; its default value @N{is 1} and can be negative.

}

@LibEntry{string.pack (fmt, v1, v2, @Cdots)|

Returns a binary string containing the values @id{v1}, @id{v2}, etc. serialized in binary form (packed) according to the format string @id{fmt} @see{pack}.

}

@LibEntry{string.packsize (fmt)|

Returns the length of a string resulting from @Lid{string.pack} with the given format.

The format string cannot have the variable-length options @Char{s} or @Char{z} @see{pack}.

}

@LibEntry{string.rep (s, n [, sep])|

Returns a string that is the concatenation of @id{n} copies of the string @id{s} separated by the string @id{sep}.

The default value for `@id{sep}` is the empty string  
(that is, no separator).  
Returns the empty string if `@id{n}` is not positive.

(Note that it is very easy to exhaust the memory of your machine  
with a single  
call to this function.)

}

@LibEntry{string.reverse (s)|

Returns a string that is the string `@id{s}` reversed.

}

@LibEntry{string.sub (s, i [, j])|

Returns the substring of `@id{s}` that  
starts at `@id{i}` and continues until `@id{j}`;  
`@id{i}` and `@id{j}` can be negative.  
If `@id{j}` is absent, then it is assumed to be equal to `@num{-1}`  
(which is the same as the string length).  
In particular,  
the call `@T{string.sub(s,1,j)}` returns a prefix of `@id{s}`  
with length `@id{j}`,  
and `@T{string.sub(s, -i)}` (for a positive `@id{i}`)  
returns a suffix of `@id{s}`  
with length `@id{i}`.

If, after the translation of negative indices,  
`@id{i}` is less than 1,  
it is corrected to 1.  
If `@id{j}` is greater than the string length,  
it is corrected to that length.  
If, after these corrections,  
`@id{i}` is greater than `@id{j}`,  
the function returns the empty string.

}

@LibEntry{string.unpack (fmt, s [, pos])|

Returns the values packed in string `@id{s}` @seeF{string.pack}  
according to the format string `@id{fmt}` @see{pack}.

An

optional `@id{pos}` marks where  
to start reading in `@id{s}` (default is 1).

After the read values,  
this function also returns the index of the first unread byte in @id{s}.

}

@LibEntry{string.upper (s)|

Receives a string and returns a copy of this string with all  
lowercase letters changed to uppercase.  
All other characters are left unchanged.  
The definition of what a lowercase letter is depends on the current locale.

}

}

@sect3{pm| @title{Patterns}

@simplesect{

Patterns in Lua are described by regular strings,  
which are interpreted as patterns by the pattern-matching functions  
@Lid{string.find},  
@Lid{string.gmatch},  
@Lid{string.gsub},  
and @Lid{string.match}.  
This section describes the syntax and the meaning  
(that is, what they match) of these strings.

}

@sect4{@title{Character Class:}

A @def{character class} is used to represent a set of characters.  
The following combinations are allowed in describing a character class:  
@description{

@item{@rep{x}|  
(where @rep{x} is not one  
of the @emphx{magic characters}  
@T{^\$()%.[]\*+-.?})  
represents the character @emph{x} itself.  
}

@item{@T{.}| (a dot) represents all characters.}

@item{@T{%a}| represents all letters.}

@item{@T{%c}| represents all control characters.}

@item{@T{%d}| represents all digits.}

@item{@T{%g}| represents all printable characters except space.}

@item{@T{%l}| represents all lowercase letters.}

@item{@T{%p}| represents all punctuation characters.}

@item{@T{%s}| represents all space characters.}

@item{@T{%u}| represents all uppercase letters.}

@item{@T{%w}| represents all alphanumeric characters.}

@item{@T{%x}| represents all hexadecimal digits.}

@item{@T{%@rep{x}}| (where @rep{x} is any non-alphanumeric character)  
represents the character @rep{x}.

This is the standard way to escape the magic characters.

Any non-alphanumeric character

(including all punctuation characters, even the non-magical)

can be preceded by a @Char{%} to represent itself in a pattern.

}

@item{@T{[@rep{set}]}|

represents

the class which is the union of all

characters in @rep{set}.

A range of characters can be specified by

separating the end characters of the range,

in ascending order, with a @Char{-}.

All classes @T{%}@emph{x} described above can also be used as  
components in @rep{set}.

All other characters in @rep{set} represent themselves.

For example, @T{[%w\_]} (or @T{[\_%w]})

represents all alphanumeric characters plus the underscore,

@T{[0-7]} represents the octal digits,

and @T{[0-7%l%-]} represents the octal digits plus  
the lowercase letters plus the @Char{-} character.

You can put a closing square bracket in a set

by positioning it as the first character in the set.

You can put a hyphen in a set

by positioning it as the first or the last character in the set.

(You can also use an escape for both cases.)

The interaction between ranges and classes is not defined.

Therefore, patterns like `@T{[%a-z]}` or `@T{[a-%]}`

have no meaning.

}

`@item{@T{[^@rep{set}]}`

represents the complement of `@rep{set}`,

where

`@rep{set}` is interpreted as above.

}

}

For all classes represented by single letters (`@T{%a}`, `@T{%c}`, etc.),

the corresponding uppercase letter represents the complement of the class.

For instance, `@T{%S}` represents all non-space characters.

The definitions of letter, space, and other character groups

depend on the current locale.

In particular, the class `@T{[a-z]}` may not be equivalent to `@T{%l}`.

}

`@sect4{@title{Pattern Item:}`

A `@def{pattern item}` can be

`@itemize{`

`@item{`

a single character class,

which matches any single character in the class;

}

`@item{`

a single character class followed by `@Char{*}`,

which matches sequences of zero or more characters in the class.

These repetition items will always match the longest possible sequence;

}

`@item{`

a single character class followed by `@Char{+}`,

which matches sequences of one or more characters in the class.

These repetition items will always match the longest possible sequence;

}

`@item{`

a single character class followed

by `@Char{-}`,

which also matches sequences of zero or more characters in the class.

Unlike `@Char{*}`,

these repetition items will always match the shortest possible sequence;

}

@item{

a single character class followed by @Char{?},

which matches zero or one occurrence of a character in the class.

It always matches one occurrence if possible;

}

@item{

@T{%@rep{n}}, for @rep{n} between 1 and 9;

such item matches a substring equal to the @rep{n}-th captured string  
(see below);

}

@item{

@T{%b@rep{xy}}, where @rep{x} and @rep{y} are two distinct characters;

such item matches strings that start @N{with @rep{x}}, end @N{with @rep{y}},

and where the @rep{x} and @rep{y} are @emph{balanced}.

This means that, if one reads the string from left to right,

counting @M{+1} for an @rep{x} and @M{-1} for a @rep{y},

the ending @rep{y} is the first @rep{y} where the count reaches 0.

For instance, the item @T{%b()} matches expressions with

balanced parentheses.

}

@item{

@T{%f[@rep{set}]}, a @def{frontier

pattern};

such item matches an empty string at any position such that

the next character belongs to @rep{set}

and the previous character does not belong to @rep{set}.

The set @rep{set} is interpreted as previously described.

The beginning and the end of the subject are handled as if

they were the character @Char{\0}.

}

}

}

@sect4{@title{Pattern:}}

A @def{pattern} is a sequence of pattern items.

A caret @Char{^} at the beginning of a pattern anchors the match at the  
beginning of the subject string.

A @Char{\$} at the end of a pattern anchors the match at the  
end of the subject string.

At other positions,

@Char{^} and @Char{\$} have no special meaning and represent themselves.

}

@sect4{@title{Captures:}}

A pattern can contain sub-patterns enclosed in parentheses;  
they describe @def{captures}.

When a match succeeds, the substrings of the subject string  
that match captures are stored (@emph{captured}) for future use.  
Captures are numbered according to their left parentheses.

For instance,

in the pattern @T{"(a\*(.)%w(%s\*))"},  
the part of the string matching @T{"a\*(.)%w(%s\*)"} is  
stored as the first capture, and therefore has @N{number 1};  
the character matching @St{.} is captured with @N{number 2},  
and the part matching @St{%s\*} has @N{number 3}.

As a special case, the capture @T{()} captures  
the current string position (a number).

For instance, if we apply the pattern @T{"()aa()"} on the  
string @T{"flaaap"}, there will be two captures: @N{3 and 5}.

}

@sect4{@title{Multiple matches:}}

The function @Lid{string.gsub} and the iterator @Lid{string.gmatch}  
match multiple occurrences of the given pattern in the subject.

For these functions,

a new match is considered valid only  
if it ends at least one byte after the end of the previous match.

In other words, the pattern machine never accepts the  
empty string as a match immediately after another match.

As an example,

consider the results of the following code:

@verbatim{

> string.gsub("abc", "()a\*()", print);

--> 1 2

--> 3

3

--> 4 4

}

The second and third results come from Lua matching an empty  
string after @Char{b} and another one after @Char{c}.

Lua does not match an empty string after @Char{a},  
because it would end at the same position of the previous match.

}



}

@sect3{pack| @title{Format Strings for Pack and Unpack}

The first argument to @Lid{string.pack},  
@Lid{string.packsize}, and @Lid{string.unpack}  
is a format string,  
which describes the layout of the structure being created or read.

A format string is a sequence of conversion options.

The conversion options are as follows:

```
@description{
@item{@T{<}|sets little endian}
@item{@T{>}|sets big endian}
@item{@T{=}|sets native endian}
@item{@T{![@rep{n}]}|sets maximum alignment to @id{n}
(default is native alignment)}
@item{@T{b}|a signed byte (@id{char})}
@item{@T{B}|an unsigned byte (@id{char})}
@item{@T{h}|a signed @id{short} (native size)}
@item{@T{H}|an unsigned @id{short} (native size)}
@item{@T{l}|a signed @id{long} (native size)}
@item{@T{L}|an
unsigned @id{long} (native size)}
@item{@T{j}|a @id{lua_Integer}}
@item{@T{J}|a @id{lua_Unsigned}}
@item{@T{T}|a @id{size_t} (native size)}
@item{@T{i[@rep{n}]}|a signed @id{int} with @id{n} bytes
(default is native size)}
@item{@T{I[@rep{n}]}|an unsigned @id{int} with @id{n} bytes
(default is native size)}
@item{@T{f}|a @id{float} (native size)}
@item{@T{d}|a @id{double} (native size)}
@item{@T{n}|a @id{lua_Number}}
@item{@T{c[@rep{n}]}|a fixed-sized string with @id{n} bytes}
@item{@T{z}|a zero-terminated string}
@item{@T{s[@emph{n}]}|a string preceded by its length
coded as an unsigned integer with @id{n} bytes
(default is a @id{size_t})}
@item{@T{x}|one byte of padding}
@item{@T{X[@rep{op}]}|an empty item that aligns
according to option @id{op}
(which is otherwise ignored)}
@item{@Char{ }|(space) ignored}
}
```

(A @St{[@rep{n}]} means an optional integral numeral.)

Except for padding, spaces, and configurations

(options @St{xX <=>!}),

each option corresponds to an argument in @Lid{string.pack}

or

a result in @Lid{string.unpack}.

For options @St{!@rep{n}}, @St{s@rep{n}}, @St{i@rep{n}}, and @St{I@rep{n}}, @id{n} can be any integer between 1 and 16.

All integral options check overflows;

@Lid{string.pack} checks whether the given value fits in the given size;

@Lid{string.unpack} checks whether the read value fits in a Lua integer.

For the unsigned options,

Lua integers are treated as unsigned values too.

Any format string starts as if prefixed by @St{!1=},

that is,

with maximum alignment of 1 (no alignment)

and native endianness.

Native endianness assumes that the whole system is either big or little endian.

The packing functions will not emulate correctly the behavior of mixed-endian formats.

Alignment works as follows:

For each option,

the format gets extra padding until the data starts

at an offset that is a multiple of the minimum between the

option size and the maximum alignment;

this minimum must be a power of 2.

Options @St{c} and @St{z} are not aligned;

option

@St{s} follows the alignment of its starting integer.

All padding is filled with zeros by @Lid{string.pack}

and ignored by @Lid{string.unpack}.

}

}

@sect2{utf8| @title{UTF-8 Support}

This library provides basic support for @x{UTF-8} encoding.

It provides all its functions inside the table @defid{utf8}.

This library does not provide any support for @x{Unicode} other than the handling of the encoding.

Any operation that needs the meaning of a character,

such as character classification, is outside its scope.

Unless stated otherwise,  
all functions that expect a byte position as a parameter  
assume that the given position is either the start of a byte sequence  
or one plus the length of the subject string.  
As in the string library,  
negative indices count from the end of the string.

Functions that create byte sequences  
accept all values up to `@T{0x7FFFFFFF}`,  
as defined in the original UTF-8 specification;  
that implies byte sequences of up to six bytes.

Functions that interpret  
byte sequences only accept  
valid sequences (well formed and not overlong).  
By default, they only accept byte sequences  
that result in valid Unicode code points,  
rejecting values greater than `@T{10FFFF}` and surrogates.  
A boolean argument `@id{lax}`, when available,  
lifts these checks,  
so that all values up to `@T{0x7FFFFFFF}` are accepted.  
(Not well formed and overlong sequences are still rejected.)

`@LibEntry{utf8.char (@Cdots)|`

Receives zero or more integers,  
converts each one to its corresponding UTF-8 byte sequence  
and returns a string with the concatenation of all these sequences.

`}`

`@LibEntry{utf8.charpattern|`

The pattern (a string, not a function) `@St{[\0-\x7F\xC2-\xFD][\x80-\xBF]*}`  
`@see{pm}`,  
which matches exactly one UTF-8 byte sequence,  
assuming that the subject is a valid UTF-8 string.

`}`

`@LibEntry{utf8.codes (s [, lax])|`

Returns values so that the construction  
`@verbatim{`  
`for p, c in utf8.codes(s) do @rep{body} end`  
`}`

will iterate over all UTF-8 characters in string `@id{s}`,  
 with  
`@id{p}` being the position (in bytes) and `@id{c}` the code point  
 of each character.  
 It raises an error if it meets any invalid byte sequence.

}

`@LibEntry{utf8.codepoint (s [, i [, j [, lax]]])|`

Returns the code points (as integers) from all characters in `@id{s}`  
 that start between byte position `@id{i}` and `@id{j}` (both included).  
 The default for `@id{i}` is 1 and for `@id{j}` is `@id{i}`.  
 It raises an error if it meets any invalid byte sequence.

}

`@LibEntry{utf8.len (s [, i [, j [, lax]]])|`

Returns the number of UTF-8 characters in string `@id{s}`  
 that start between positions `@id{i}` and `@id{j}` (both inclusive).  
 The default for `@id{i}` is `@num{1}` and for `@id{j}` is `@num{-1}`.  
 If it finds any invalid byte sequence,  
 returns `@fail` plus the position of the first invalid byte.

}

`@LibEntry{utf8.offset (s, n [, i])|`

Returns the position (in bytes) where the encoding of the  
`@id{n}`-th character of `@id{s}`  
 (counting from position `@id{i}`) starts.  
 A negative `@id{n}` gets characters before position `@id{i}`.  
 The  
 default for `@id{i}` is 1 when `@id{n}` is non-negative  
 and `@T{#s + 1}` otherwise,  
 so that `@T{utf8.offset(s, -n)}` gets the offset of the  
`@id{n}`-th character from the end of the string.  
 If the specified character is neither in the subject  
 nor right after its end,  
 the function returns `@fail`.

As a special case,  
 when `@id{n}` is 0 the function returns the start of the encoding  
 of the character that contains the `@id{i}`-th byte of `@id{s}`.

This function assumes that `@id{s}` is a valid UTF-8 string.

```
}
```

```
}
```

@sect2{tablib| @title{Table Manipulation}

This library provides generic functions for table manipulation.  
It provides all its functions inside the table @defid{table}.

Remember that, whenever an operation needs the length of a table,  
all caveats about the length operator apply @see{len-op}.  
All functions ignore non-numeric keys  
in the tables given as arguments.

@LibEntry{table.concat (list [, sep [, i [, j]])|

Given a list where all elements are strings or numbers,  
returns the string

@T{list[i]..sep..list[i+1] @Cdots sep..list[j]}.

The default value for @id{sep} is the empty string,  
the default for @id{i} is 1,  
and the default for @id{j} is @T{#list}.

If @id{i} is greater than @id{j}, returns the empty string.

```
}
```

@LibEntry{table.insert (list, [pos,] value)|

Inserts element @id{value} at position @id{pos} in @id{list},  
shifting up the elements

@T{list[pos], list[pos+1], @Cdots, list[#list]}.

The default value for @id{pos} is @T{#list+1},  
so that a call @T{table.insert(t,x)} inserts @id{x} at the end  
of the list @id{t}.

```
}
```

@LibEntry{table.move (a1, f, e, t [,a2])|

Moves elements from the table @id{a1} to the table @id{a2},  
performing the equivalent to the following  
multiple assignment:

@T{a2[t],@Cdots = a1[f],@Cdots,a1[e]}.

The default for @id{a2} is @id{a1}.

The destination range can overlap with the source range.

The number of elements to be moved must fit in a Lua integer.

Returns the destination table `@id{a2}`.

}

`@LibEntry{table.pack (@Cdots)|`

Returns

a new table with all arguments stored into keys 1, 2, etc.  
and with a field `@St{n}` with the total number of arguments.  
Note that the resulting table may not be a sequence,  
if some arguments are `@nil`.

}

`@LibEntry{table.remove (list [, pos])|`

Removes from `@id{list}` the element at position `@id{pos}`,  
returning the value of the removed element.

When `@id{pos}` is an integer between 1 and `@T{#list}`,  
it shifts down the elements

`@T{list[pos+1], list[pos+2], @Cdots, list[#list]}`  
and erases element `@T{list[#list]}`;

The index `@id{pos}` can also be 0 when `@T{#list}` is 0,  
or `@T{#list + 1}`.

The default value for `@id{pos}` is `@T{#list}`,  
so that a call `@T{table.remove(1)}` removes the last element  
of the list `@id{l}`.

}

`@LibEntry{table.sort (list [, comp])|`

Sorts the list elements in a given order, *@emph{in-place}*,  
from `@T{list[1]}` to `@T{list[#list]}`.

If `@id{comp}` is given,

then it must be a function that receives two list elements  
and returns true when the first element must come  
before the second in

the final order,

so that, after the sort,

`@T{i <= j}` implies `@T{not comp(list[j],list[i])}`.

If `@id{comp}` is not given,

then the standard Lua operator `@T{<}` is used instead.

The `@id{comp}` function must define a consistent order;  
more formally, the function must define a strict weak order.  
(A weak order is similar to a total order,

but it can equate different elements for comparison purposes.)

The sort algorithm is not stable:

Different elements considered equal by the given order  
may have their relative positions changed by the sort.

}

@LibEntry{table.unpack (list [, i [, j]])|

Returns the elements from the given list.

This function is equivalent to

```
@verbatim{
return list[i], list[i+1], @Cdots, list[j]
}
```

By default, @id{i} @N{is 1} and @id{j} is @T{#list}.

}

}

@sect2{mathlib| @title{Mathematical Functions}}

This library provides basic mathematical functions.

It provides all its functions and constants inside the table @defid{math}.

Functions with the annotation @St{integer/float}

give

integer results for integer arguments

and float results for non-integer arguments.

The rounding functions

@Lid{math.ceil}, @Lid{math.floor}, and @Lid{math.modf}

return an integer when the result fits in the range of an integer,  
or a float otherwise.

@LibEntry{math.abs (x)|

Returns the maximum value between @id{x} and @id{-x}. (integer/float)

}

@LibEntry{math.acos (x)|

Returns the arc cosine of @id{x} (in radians).

}

@LibEntry{math.asin (x)|

Returns the arc sine of  $\text{@id}\{x\}$  (in radians).

}

`@LibEntry{math.atan (y [, x])|`

`@index{atan} @index{atan2}`

Returns the arc tangent of  $\text{@T}\{y/x\}$  (in radians),  
using the signs of both arguments to find the  
quadrant of the result.

It also handles correctly the case of  $\text{@id}\{x\}$  being zero.

The default value for  $\text{@id}\{x\}$  is 1,  
so that the call `@T{math.atan(y)}`  
returns the arc tangent of  $\text{@id}\{y\}$ .

}

`@LibEntry{math.ceil (x)|`

Returns the smallest integral value greater than or equal to  $\text{@id}\{x\}$ .

}

`@LibEntry{math.cos (x)|`

Returns the cosine of  
 $\text{@id}\{x\}$  (assumed to be in radians).

}

`@LibEntry{math.deg (x)|`

Converts the angle  $\text{@id}\{x\}$  from radians to degrees.

}

`@LibEntry{math.exp (x)|`

Returns the value  $\text{@M}\{e^{\text{@sp}\{x\}}\}$   
(where  $\text{@id}\{e\}$  is the base of natural logarithms).

}

`@LibEntry{math.floor (x)|`

Returns the largest integral value less than or equal to  $\text{@id}\{x\}$ .



}

@LibEntry{math.fmod (x, y)|

Returns the remainder of the division of @id{x} by @id{y} that rounds the quotient towards zero. (integer/float)

}

@LibEntry{math.huge|

The float value @idx{HUGE\_VAL},  
a value greater than any other numeric value.

}

@LibEntry{math.log (x [, base])|

Returns the logarithm of @id{x} in the given base.  
The default for @id{base} is @M{e}  
(so that the function returns the natural logarithm of @id{x}).

}

@LibEntry{math.max (x, @Cdots)|

Returns the argument with the maximum value,  
according to the Lua operator @T{<}.

}

@LibEntry{math.maxinteger|

An integer with the maximum value for an integer.

}

@LibEntry{math.min (x, @Cdots)|

Returns  
the argument with the minimum value,  
according to the Lua operator @T{<}.

}

@LibEntry{math.mininteger|

An integer with the minimum value for an integer.

}

@LibEntry{math.modf (x)|

Returns the integral part of @id{x} and the fractional part of @id{x}.  
Its second result is always a float.

}

@LibEntry{math.pi|

The value of @M{@pi}.

}

@LibEntry{math.rad (x)|

Converts the angle @id{x} from degrees to radians.

}

@LibEntry{math.random ([m [, n]])|

When called without arguments,  
returns a pseudo-random float with uniform distribution  
in the range @C{() @M{[0,1)}. @C{[]}

When called with two integers @id{m} and @id{n},  
@id{math.random} returns a pseudo-random integer  
with uniform distribution in the range @M{[m, n]}.

The call @T{math.random(n)}, for a positive @id{n},  
is equivalent to @T{math.random(1,n)}.

The call @T{math.random(0)} produces an integer with  
all bits (pseudo)random.

This function uses the @idx{xoshiro256\*\*} algorithm to produce  
pseudo-random 64-bit integers,  
which

are the results of calls with @N{argument 0}.

Other results (ranges and floats)

are unbiased extracted from these integers.

Lua initializes its pseudo-random generator with the equivalent of  
a call to @Lid{math.randomseed} with no arguments,  
so that @id{math.random} should generate  
different sequences of results each time the program runs.

}

@LibEntry{math.randomseed ([x [, y]])|

When called with at least one argument,  
the integer parameters `@id{x}` and `@id{y}` are  
joined into a 128-bit `@emphx{seed}` that  
is used to reinitialize the pseudo-random generator;  
equal seeds produce equal sequences of numbers.  
The default for `@id{y}` is zero.

When called with no arguments,  
Lua generates a seed with  
a weak attempt for randomness.

This function returns the two seed components  
that were effectively used,  
so that setting them again repeats the sequence.

To ensure a required level of randomness to the initial state  
(or contrarily, to have a deterministic sequence,  
for instance when debugging  
a program),  
you should call `@Lid{math.randomseed}` with explicit arguments.

}

`@LibEntry{math.sin (x)|`

Returns the sine of `@id{x}` (assumed to be in radians).

}

`@LibEntry{math.sqrt (x)|`

Returns the square root of `@id{x}`.  
(You can also use the expression `@T{x^0.5}` to compute this value.)

}

`@LibEntry{math.tan (x)|`

Returns the tangent of `@id{x}` (assumed to be in radians).

}

`@LibEntry{math.tointeger (x)|`

If the value `@id{x}` is convertible to an integer,  
returns that integer.  
Otherwise, returns `@fail`.

}

@LibEntry{math.type (x)|

Returns @St{integer} if @id{x} is an integer,  
@St{float} if it is a float,  
or @fail if @id{x} is not a number.

}

@LibEntry{math.ult (m, n)|

Returns a boolean,  
@true if and only if integer @id{m} is below integer @id{n} when  
they are compared as @x{unsigned integers}.

}

}

@sect2{iolib| @title{Input and Output Facilities}}

The I/O library provides two different styles for file manipulation.  
The first one uses implicit file handles;  
that is, there  
are operations to set a default input file and a  
default output file,  
and all input/output operations are done over these default files.  
The second style uses explicit file handles.

When using implicit file handles,  
all operations are supplied by table @defid{io}.  
When using explicit file handles,  
the operation @Lid{io.open} returns a file handle  
and then all operations are supplied as methods of the file handle.

The metatable for file handles provides metamethods  
for @idx{\_\_gc} and @idx{\_\_close} that try  
to close the file when called.

The table @id{io} also provides  
three predefined file handles with their usual meanings from C:  
@defid{io.stdin}, @defid{io.stdout}, and @defid{io.stderr}.  
The I/O library never closes these files.

Unless otherwise stated,  
all I/O functions return @fail on failure,

plus an error message as a second result and  
a system-dependent error code as a third result,  
and some non-false value on success.  
On non-POSIX systems,  
the computation of the error message  
and error code  
in case of errors  
may be not `@x{thread safe}`,  
because they rely on the global C variable `@id{errno}`.

`@LibEntry{io.close ([file])|`

Equivalent to `@T{file:close()}`.  
Without a `@id{file}`, closes the default output file.

`}`

`@LibEntry{io.flush ()|`

Equivalent to `@T{io.output():flush()}`.

`}`

`@LibEntry{io.input ([file])|`

When called with a file name, it opens the named file (in text mode),  
and sets its handle as the default input file.

When called with a file handle,  
it simply sets this file handle as the default input file.

When called without arguments,  
it returns the current default input file.

In case of errors this function raises the error,  
instead of returning an error code.

`}`

`@LibEntry{io.lines ([filename, @Cdots])|`

Opens the given file name in read mode  
and returns an iterator function that  
works like `@T{file:lines(@Cdots)}` over the opened file.

When the iterator function fails to read any value,  
it automatically closes the file.

Besides the iterator function,

`@id{io.lines}`

returns three other values:  
two `@nil` values as placeholders,

plus the created file handle.

Therefore, when used in a generic `@Rw{for}` loop, the file is closed also if the loop is interrupted by an error or a `@Rw{break}`.

The call `@T{io.lines()}` (with no file name) is equivalent to `@T{io.input():lines("")}`;

that is, it iterates over the lines of the default input file.

In this case, the iterator does not close the file when the loop ends.

In case of errors opening the file, this function raises the error, instead of returning an error code.

```
}
```

```
@LibEntry{io.open (filename [, mode])|
```

This function opens a file, in the mode specified in the string `@id{mode}`.

In case of success, it returns a new file handle.

The `@id{mode}` string can be any of the following:

```
@description{
```

```
@item{@St{r}| read mode (the default);}
```

```
@item{@St{w}| write mode;}
```

```
@item{@St{a}| append mode;}
```

```
@item{@St{r+}| update mode, all previous data is preserved;}
```

```
@item{@St{w+}| update mode, all previous data is  
erased;}
```

```
@item{@St{a+}| append update mode, previous data is preserved,  
writing is only allowed at the end of file.}
```

```
}
```

The `@id{mode}` string can also have a `@Char{b}` at the end, which is needed in some systems to open the file in binary mode.

```
}
```

```
@LibEntry{io.output ([file])|
```

Similar to `@Lid{io.input}`, but operates over the default output file.

```
}
```

```
@LibEntry{io.popen (prog [, mode])|
```

This function is system dependent and is not available

on all platforms.

Starts the program `@id{prog}` in a separated process and returns a file handle that you can use to read data from this program (if `@id{mode}` is `@T{"r"}`, the default) or to write data to this program (if `@id{mode}` is `@T{"w"}`).

}

`@LibEntry{io.read (@Cdots)|`

Equivalent to `@T{io.input():read(@Cdots)}`.

}

`@LibEntry{io.tmpfile ()|`

In case of success,  
returns a handle for a temporary file.  
This file is opened in update mode  
and it is automatically removed when the program ends.

}

`@LibEntry{io.type (obj)|`

Checks whether `@id{obj}`  
is a valid file handle.  
Returns the string `@T{"file"}` if `@id{obj}` is an open file handle,  
`@T{"closed file"}` if `@id{obj}` is a closed file handle,  
or `@fail` if `@id{obj}` is not a file handle.

}

`@LibEntry{io.write (@Cdots)|`

Equivalent to `@T{io.output():write(@Cdots)}`.

}

`@LibEntry{file:close ()|`

Closes `@id{file}`.  
Note that files are automatically closed when  
their handles are garbage collected,  
but that takes an unpredictable amount of time to happen.

When closing a file handle created with `@Lid{io.popen}`,  
`@Lid{file:close}` returns the same values  
returned by `@Lid{os.execute}`.

```
}
```

```
@LibEntry{file:flush ()|
```

Saves any written data to `@id{file}`.

```
}
```

```
@LibEntry{file:lines (@Cdots)|
```

Returns an iterator function that,  
each time it is called,  
reads the file according to the given formats.

When no format is given,  
uses `@St{1}` as a default.

As an example, the construction

```
@verbatim{  
for c in file:lines(1) do @rep{body} end  
}
```

will iterate over all characters of the file,  
starting at the current  
position.

Unlike `@Lid{io.lines}`, this function does not close the file  
when the loop ends.

```
}
```

```
@LibEntry{file:read (@Cdots)|
```

Reads the file `@id{file}`,  
according to the given formats, which specify what to read.

For each format,  
the function returns a string or a number with the characters read,  
or `@fail` if it cannot read data with the specified format.

(In this latter case,  
the function does not read subsequent formats.)

When called without arguments,  
it uses a default format that reads the next line  
(see below).

The available formats are

```
@description{
```



`@item{@St{n}|`  
 reads a numeral and returns it as a float or an integer,  
 following the lexical conventions of Lua.  
 (The numeral may have leading whitespaces and a sign.)  
 This format always reads the longest input sequence that  
 is a valid prefix for a numeral;  
 if that prefix does not form a valid numeral  
 (e.g., an empty string, `@St{0x}`, or `@St{3.4e-}`)  
 or it is too long (more than 200 characters),  
 it is discarded and the format returns `@fail`.  
`}`

`@item{@St{a}|`  
 reads  
 the whole file, starting at the current position.  
 On end of file, it returns the empty string;  
 this format never fails.  
`}`

`@item{@St{l}|`  
 reads the next line skipping the end of line,  
 returning `@fail` on end of file.  
 This is the default format.  
`}`

`@item{@St{L}|`  
 reads the next line keeping the end-of-line character (if present),  
 returning `@fail` on end of file.  
`}`

`@item{@emph{number}|`  
 reads a string with up to this number of bytes,  
 returning `@fail` on end of file.  
 If `@id{number}` is zero,  
 it reads nothing and returns an empty string,  
 or `@fail` on end of file.  
`}`

`}`  
 The formats `@St{l}` and `@St{L}` should be used only for text files.  
`}`

`@LibEntry{file:seek ([whence [, offset]])}`

Sets and gets the file position,  
 measured from the beginning of the file,  
 to the position given by `@id{offset}` plus a base

specified by the string @id{whence}, as follows:

```
@description{
@item{@St{set}| base is position 0 (beginning of the file);}
@item{@St{cur}| base is current position;}
@item{@St{end}|
base is end of file;}
}
```

In case of success, @id{seek} returns the final file position, measured in bytes from the beginning of the file.

If @id{seek} fails, it returns @fail, plus a string describing the error.

The default value for @id{whence} is @T{"cur"}, and for @id{offset} is 0.

Therefore, the call @T{file:seek()} returns the current file position, without changing it; the call @T{file:seek("set")} sets the position to the beginning of the file (and returns 0); and the call @T{file:seek("end")} sets the position to the end of the file, and returns its size.

```
}
```

```
@LibEntry{file:setvbuf (mode [, size])|
```

Sets the buffering mode for a file.

There are three available modes:

```
@description{
@item{@St{no}| no buffering.}
@item{@St{full}| full buffering.}
@item{@St{line}| line buffering.}
}
```

For the last two cases,

@id{size} is a hint for the size of the buffer, in bytes.

The default is an appropriate size.

The specific behavior of each mode is non portable; check the underlying

@ANSI{setvbuf} in your platform for more details.

```
}
```

```
@LibEntry{file:write (@Cdots)|
```

Writes the value of each of its arguments to @id{file}.

The arguments must be strings or numbers.

In case of success, this function returns @id{file}.

}

}

@sect2{oslib| @title{Operating System Facilities}}

This library is implemented through table @defid{os}.

@LibEntry{os.clock ()|

Returns an approximation of the amount in seconds of CPU time  
used by the program,  
as returned by the underlying @ANSI{clock}.

}

@LibEntry{os.date ([format [, time]])|

Returns a string or a table containing date and time,  
formatted according to the given string @id{format}.

If the @id{time} argument is present,  
this is the time to be formatted  
(see the @Lid{os.time} function for a description of this value).  
Otherwise, @id{date} formats the current time.

If @id{format} starts with @Char{!},  
then the date is formatted in Coordinated Universal Time.  
After this optional character,  
if @id{format} is the string @St{\*t},  
then

@id{date} returns a table with the following fields:  
@id{year}, @id{month} (1@En{ }12), @id{day} (1@En{ }31),  
@id{hour} (0@En{ }23), @id{min} (0@En{ }59),  
@id{sec} (0@En{ }61, due to leap seconds),  
@id{yday} (weekday, 1@En{ }7, Sunday @N{is 1}),  
@id{yday} (day of the year, 1@En{ }366),  
and @id{isdst} (daylight saving flag, a boolean).  
This last field may be absent  
if the information is not available.

If @id{format} is not @St{\*t},  
then @id{date} returns the date as a string,  
formatted according to the same rules as the @ANSI{strftime}.

If `@id{format}` is absent, it defaults to `@St{%c}`,  
which gives a human-readable date and time representation  
using the current locale.

On non-POSIX systems,  
this function may be not `@x{thread safe}`  
because of its reliance on `@CId{gmtime}` and `@CId{localtime}`.

}

`@LibEntry{os.difftime (t2, t1)|`

Returns the difference, in seconds,  
from time `@id{t1}` to time `@id{t2}`  
(where the times are values returned by `@Lid{os.time}`).  
In `@x{POSIX}`, `@x{Windows}`, and  
some other systems,  
this value is exactly `@id{t2}@M{-}@id{t1}`.

}

`@LibEntry{os.execute ([command])|`

This function is equivalent to the `@ANSI{system}`.  
It passes `@id{command}` to be executed by an operating system shell.  
Its first result is `@true`  
if the command terminated successfully,  
or `@fail` otherwise.  
After this first result  
the function returns a string plus a number,  
as follows:  
`@description{`

`@item{@St{exit}|`  
the command terminated normally;  
the following number is the exit status of the command.  
}

`@item{@St{signal}|`  
the command was terminated by a signal;  
the following number is the signal that terminated the command.  
}

}

When called without a `@id{command}`,  
`@id{os.execute}` returns a boolean that is true if a shell is available.

}

@LibEntry{os.exit ([code [, close]])|

Calls the @ANSI{exit} to terminate the host program.

If @id{code} is @true,

the returned status is @idx{EXIT\_SUCCESS};

if @id{code} is @false,

the returned status is @idx{EXIT\_FAILURE};

if @id{code} is a number,

the

returned status is this number.

The default value for @id{code} is @true.

If the optional second argument @id{close} is true,

the function closes the Lua state before exiting @seeF{lua\_close}.

}

@LibEntry{os.getenv (varname)|

Returns the value of the process environment variable @id{varname}

or @fail if the variable is not defined.

}

@LibEntry{os.remove (filename)|

Deletes the file (or empty directory, on @x{POSIX} systems)  
with the given name.

If this function fails, it returns @fail

plus a string describing the error and the error code.

Otherwise, it returns true.

}

@LibEntry{os.rename (oldname, newname)|

Renames the file or directory named @id{oldname} to @id{newname}.

If this function fails, it returns @fail,

plus a string describing the error and the error code.

Otherwise, it returns true.

}

@LibEntry{os.setlocale (locale [, category])|

Sets the current locale of the program.

@id{locale} is a system-dependent string specifying a locale;

@id{category} is an optional

string describing which category to change:

@T{"all"}, @T{"collate"}, @T{"ctype"},

@T{"monetary"}, @T{"numeric"}, or @T{"time"};

the default category is @T{"all"}.

The function returns the name of the new locale,

or @fail if the request cannot be honored.

If @id{locale} is the empty string,

the current locale is set to an implementation-defined native locale.

If @id{locale} is the string @St{C},

the current locale is set to the standard C locale.

When called with @nil as the first argument,

this function only returns the name of the current locale

for the given category.

This function may be not @x{thread safe}

because of its reliance on @Cid{setlocale}.

}

@LibEntry{os.time ([table])|

Returns the current time when called without arguments,

or a time representing the local date and time specified by the given table.

This table must have fields @id{year}, @id{month}, and @id{day},

and may have fields

@id{hour} (default is 12),

@id{min} (default is 0),

@id{sec} (default is 0),

and

@id{isdst} (default is @nil).

Other fields are ignored.

For a description of these fields, see the @Lid{os.date} function.

When the function is called,

the values in these fields do not need to be inside their valid ranges.

For instance, if @id{sec} is -10,

it means 10 seconds before the time specified by the other fields;

if @id{hour} is 1000,

it means 1000 hours after the time specified by the other fields.

The returned value is a number, whose meaning depends on your system.

In @x{POSIX}, @x{Windows}, and some other systems,

this number counts the number

of seconds since some given start time (the `@Q{epoch}`).

In other systems, the meaning is not specified,  
and the number returned by `@id{time}` can be used only as an argument to  
`@Lid{os.date}` and `@Lid{os.difftime}`.

When called with a table,  
`@id{os.time}` also normalizes all the fields  
documented in the `@Lid{os.date}` function,  
so that they represent the same time as before the call  
but with values inside their valid ranges.

}

`@LibEntry{os.tmpname`  
`()`

Returns a string with a file name that can  
be used for a temporary file.  
The file must be explicitly opened before its use  
and explicitly removed when no longer needed.

In `@x{POSIX}` systems,  
this function also creates a file with that name,  
to avoid security risks.  
(Someone else might create the file with wrong permissions  
in the time between getting the name and creating the file.)  
You still have to open the file to use it  
and to remove it (even if you do not use it).

When possible,  
you may prefer to use `@Lid{io.tmpfile}`,  
which automatically removes the file when the program ends.

}

}

`@sect2{debuglib| @title{The Debug Library}`

This library provides  
the functionality of the `@link{debugI|debug interface}` to Lua programs.  
You should exert care when using this library.  
Several of its functions  
violate basic assumptions about Lua code  
(e.g., that variables local to a function  
cannot be accessed from outside;  
that userdata metatables cannot be changed by Lua

code;  
that Lua programs do not crash)  
and therefore can compromise otherwise secure code.  
Moreover, some functions in this library may be slow.

All functions in this library are provided  
inside the `@defid{debug}` table.  
All functions that operate over a thread  
have an optional first argument which is the  
thread to operate over.  
The default is always the current thread.

`@LibEntry{debug.debug ()|`

Enters an interactive mode with the user,  
running each string that the user enters.  
Using simple commands and other debug facilities,  
the user can inspect global and local variables,  
change their values, evaluate expressions, and so on.  
A line containing only the word `@id{cont}` finishes this function,  
so that the caller continues its execution.

Note that commands for `@id{debug.debug}` are not lexically nested  
within any function and so have no direct access to local variables.

`}`

`@LibEntry{debug.gethook ([thread])|`

Returns the current hook settings of the thread, as three values:  
the current  
hook function, the current hook mask,  
and the current hook count,  
as set by the `@Lid{debug.sethook}` function.

Returns `@fail` if there is no active hook.

`}`

`@LibEntry{debug.getinfo ([thread,] f [, what])|`

Returns a table with information about a function.  
You can give the function directly  
or you can give a number as the value of `@id{f}`,  
which means the function running at level `@id{f}` of the call stack  
of the given thread:  
`@N{level 0}` is the current function (`@id{getinfo}` itself);



@N{level 1} is the function that called @id{getinfo}  
(except for tail calls, which do not count in the stack);  
and so on.

If @id{f} is a number greater than the number of active functions,  
then @id{getinfo} returns @fail.

The returned table can contain all the fields returned by @Lid{lua\_getinfo},  
with the string @id{what} describing which fields to fill in.

The default for @id{what} is to get all information available,  
except the table of valid lines.

The option @Char{f}  
adds a field named @id{func} with  
the function itself.

The option @Char{L} adds a field named @id{activelines}  
with the table of valid lines,  
provided the function is a Lua function.  
If the function has no debug information,  
the table is empty.

For instance, the expression @T{debug.getinfo(1,"n").name} returns  
a name for the current function,  
if a reasonable name can be found,  
and the expression @T{debug.getinfo(print)}  
returns a table with all available information  
about the @Lid{print} function.

}

@LibEntry{debug.getlocal ([thread,] f, local)|

This function returns the name and the value of the local variable  
with index @id{local} of the function at level @id{f} of the stack.  
This function accesses not only explicit local variables,  
but also parameters and temporary values.

The first parameter or local variable has @N{index 1}, and so on,  
following the order that they are declared in the code,  
counting only the variables that are active  
in the current scope of the function.

Compile-time constants may not appear  
in this listing,

if they were optimized away by the compiler.

Negative indices refer to vararg arguments;

@num{-1} is the first vararg argument.

The function returns @fail

if there is no variable with the given index,

and raises an error when called with a level out of range.

(You can call @Lid{debug.getinfo} to check whether the level is valid.)

Variable names starting with @Char{() (open parenthesis) @C{}} represent variables with no known names (internal variables such as loop control variables, and variables from chunks saved without debug information).

The parameter @id{f} may also be a function. In that case, @id{getlocal} returns only the name of function parameters.

}

@LibEntry{debug.getmetatable (value)|

Returns the metatable of the given @id{value} or @nil if it does not have a metatable.

}

@LibEntry{debug.getregistry ()|

Returns the registry table @see{registry}.

}

@LibEntry{debug.getupvalue (f, up)|

This function returns the name and the value of the upvalue with

index @id{up} of the function @id{f}.

The function returns @fail if there is no upvalue with the given index.

(For Lua functions, upvalues are the external local variables that the function uses, and that are consequently included in its closure.)

For @N{C functions}, this function uses the empty string @T{""} as a name for all upvalues.

Variable name @Char{?} (interrogation mark) represents variables with no known names (variables from chunks saved without debug information).

}

@LibEntry{debug.getuservalue (u, n)|

Returns the @id{n}-th user value associated

to the userdata `@id{u}` plus a boolean,  
`@false` if the userdata does not have that value.

}

`@LibEntry{debug.sethook ([thread,] hook, mask [, count])|`

Sets the given function as the debug hook.

The string `@id{mask}` and the number `@id{count}` describe  
when the hook will be called.

The string mask may have any combination of the following characters,  
with the given meaning:

`@description{`

`@item{@Char{c}|` the hook is called every  
time Lua calls a function;

`@item{@Char{r}|` the hook is called every time Lua returns from a function;

`@item{@Char{l}|` the hook is called every time Lua enters a new line of code.)

}

Moreover,

with a `@id{count}` different from zero,  
the hook is called also after every `@id{count}` instructions.

When called without arguments,

`@Lid{debug.sethook}` turns off the hook.

When the hook is called, its first parameter is a string  
describing the event that has triggered its call:

`@T{"call"}`, `@T{"tail call"}`, `@T{"return"}`,

`@T{"line"}`, and `@T{"count"}`.

For line events,

the hook also gets the new line number as its second parameter.

Inside a hook,

you can call `@id{getinfo}` with `@N{level 2}` to get more information about  
the running function.

(`@N{Level 0}` is the `@id{getinfo}` function,

and `@N{level 1}` is the hook function.)

}

`@LibEntry{debug.setlocal ([thread,] level, local, value)|`

This function assigns the value `@id{value}` to the local variable  
with index `@id{local}` of the function at level `@id{level}`  
of the stack.

The function returns `@fail` if there is no local  
variable with the given index,

and raises an error when called with a `@id{level}` out of range.

(You can call `@id{getinfo}` to check whether the level is valid.)

Otherwise, it returns the name of the local variable.

See @Lid{debug.getlocal} for more information about variable indices and names.

}

@LibEntry{debug.setmetatable (value, table)|

Sets the metatable for the given @id{value} to the given @id{table} (which can be @nil).

Returns @id{value}.

}

@LibEntry{debug.setupvalue (f, up, value)|

This function assigns the value @id{value} to the upvalue with index @id{up} of the function @id{f}.

The function returns @fail if there is no upvalue with the given index.

Otherwise, it returns the name of the upvalue.

See @Lid{debug.getupvalue} for more information about upvalues.

}

@LibEntry{debug.setuservalue (udata, value, n)|

Sets the given @id{value} as the @id{n}-th user value associated to the given @id{udata}.

@id{udata}

must be a full userdata.

Returns @id{udata},

or @fail if the userdata does not have that value.

}

@LibEntry{debug.traceback ([thread,] [message [, level]])|

If @id{message} is present but is neither a string nor @nil, this function returns @id{message} without further processing.

Otherwise,

it returns a string with a traceback of the call stack.

The optional @id{message} string is appended at the beginning of the traceback.

An optional @id{level} number tells at which level

to start the traceback  
(default is 1, the function calling @id{traceback}).

}

@LibEntry{debug.upvalueid (f, n)|

Returns a unique identifier (as a light userdata)  
for the upvalue numbered @id{n}  
from the given function.

These unique identifiers allow a program to check whether different  
closures share upvalues.

Lua closures that share an upvalue  
(that is, that access a same external local variable)  
will return identical ids for those upvalue indices.

}

@LibEntry{debug.upvaluejoin (f1, n1, f2, n2)|

Make  
the @id{n1}-th upvalue of the Lua closure @id{f1}  
refer to the @id{n2}-th upvalue of the Lua closure @id{f2}.

}

}

}

@C{-----}  
@sect1{lua-sa| @title{Lua Standalone}}

Although Lua has been designed as an extension language,  
to be embedded in a host @N{C program},  
it is also frequently used as a standalone language.  
An interpreter for Lua as a standalone language,  
called simply @id{lua},  
is provided with the standard distribution.  
The @x{standalone interpreter} includes  
all standard libraries.

Its usage is:

@verbatim{  
lua [options] [script [args]]  
}

The options are:

```

@description{
@item{@T{-e @rep{stat}}| execute string @rep{stat};}
@item{@T{-i}| enter interactive mode after running @rep{script};}
@item{@T{-l @rep{mod}}| @Q{require} @rep{mod} and assign the
result to global @rep{mod};}
@item{@T{-l @rep{g=mod}}| @Q{require} @rep{mod} and assign the
result to global @rep{g};}
@item{@T{-v}| print version information;}
@item{@T{-E}|
ignore environment variables;}
@item{@T{-W}| turn warnings on;}
@item{@T{--}| stop handling options;}
@item{@T{-}| execute @id{stdin} as a file and stop handling options.}
}
(The form @T{-l @rep{g=mod}} was introduced in @N{release 5.4.4}.)

```

After handling its options, @id{lua} runs the given @emph{script}.

When called without arguments,

@id{lua} behaves as @T{lua -v -i}

when the standard input (@id{stdin}) is a terminal,

and as @T{lua -} otherwise.

When called without the option @T{-E},

the interpreter checks for an environment variable @defid{LUA\_INIT\_5\_4}

(or @defid{LUA\_INIT} if the versioned name is not defined)

before running any argument.

If the variable content has the format @T{@At@rep{filename}},

then @id{lua} executes the file.

Otherwise, @id{lua} executes the string itself.

When called with the option @T{-E},

Lua does not consult any environment variables.

In particular,

the values of @Lid{package.path} and @Lid{package.cpath}

are set with the default

paths defined in @id{luaconf.h}.

To signal to the libraries that this option is on,

the stand-alone interpreter sets the field

@idx{"LUA\_NOENV"} in the registry to a true value.

Other libraries may consult this field for the same purpose.

The options @T{-e}, @T{-l}, and @T{-W} are handled in the order they appear.

For instance, an invocation like

```

@verbatim{
$ lua -e 'a=1' -llib1 script.lua
}

```

will first set @id{a} to 1, then require the library @id{lib1},

and finally run the file `@id{script.lua}` with no arguments.  
(Here `@T{$}` is the shell prompt. Your prompt may be different.)

Before running any code,  
`@id{lua}` collects all command-line arguments  
in a global table called `@id{arg}`.  
The script name goes to index 0,  
the first argument after the script name goes to index 1,  
and so on.

Any arguments before the script name  
(that is, the interpreter name plus its options)  
go to negative indices.

For instance, in the call

```
@verbatim{
$ lua -la b.lua t1 t2
}
```

the table is like this:

```
@verbatim{
arg
= { [-2] = "lua", [-1] = "-la",
    [0] = "b.lua",
    [1] = "t1", [2] = "t2" }
}
```

If there is no script in the call,  
the interpreter name goes to index 0,  
followed by the other arguments.

For instance, the call

```
@verbatim{
$ lua -e "print(arg[1])"
}
```

will print `@St{-e}`.

If there is a script,  
the script is called with arguments  
`@T{arg[1]}`, `@Cdots`, `@T{arg[#arg]}`.

Like all chunks in Lua,  
the script is compiled as a variadic function.

In interactive mode,  
Lua repeatedly prompts and waits for a line.  
After reading a line,  
Lua first try to interpret the line as an expression.  
If it succeeds, it prints its value.  
Otherwise, it interprets the line as a statement.  
If you write an incomplete statement,  
the interpreter waits for its completion  
by issuing a different prompt.

If the global variable `@defid{_PROMPT}` contains a string,

then its value is used as the prompt.  
Similarly, if the global variable `@defid{_PROMPT2}` contains a string,  
its value is used as the secondary prompt  
(issued  
during incomplete statements).

In case of unprotected errors in the script,  
the interpreter reports the error to the standard error stream.  
If the error object is not a string but  
has a metamethod `@idx{__tostring}`,  
the interpreter calls this metamethod to produce the final message.  
Otherwise, the interpreter converts the error object to a string  
and adds a stack traceback to it.  
When warnings are on,  
they are simply printed in the standard error output.

When finishing normally,  
the interpreter closes its main Lua state  
`@seeF{lua_close}`.  
The script can avoid this step by  
calling `@Lid{os.exit}` to terminate.

To allow the use of Lua as a  
script interpreter in Unix systems,  
Lua skips the first line of a file chunk if it starts with `@T{#}`.  
Therefore, Lua scripts can be made into executable programs  
by using `@T{chmod +x}` and `@N{the @T{#!}}` form,  
as in

```
@verbatim{
#!/usr/local/bin/lua
}
```

Of course,  
the location of the Lua interpreter may be different in your machine.

If `@id{lua}` is in your

`@id{PATH}`,

then

```
@verbatim{
#!/usr/bin/env lua
}
```

is a more portable solution.

```
}
```

`@sect1{incompat| @title{Incompatibilities with the Previous Version}`

`@simplesect{`



Here we list the incompatibilities that you may find when moving a program from @N{Lua 5.3} to @N{Lua 5.4}.

You can avoid some incompatibilities by compiling Lua with appropriate options (see file @id{luaconf.h}).

However,

all these compatibility options will be removed in the future.

More often than not,

compatibility issues arise when these compatibility options are removed.

So, whenever you have the chance,

you should try to test your code with a version of Lua compiled with all compatibility options turned off.

That will ease transitions to newer versions of Lua.

Lua versions can always change the C API in ways that do not imply source-code changes in a program, such as the numeric values for constants or the implementation of functions as macros.

Therefore,

you should never assume that binaries are compatible between different

Lua versions.

Always recompile clients of the Lua API when using a new version.

Similarly, Lua versions can always change the internal representation of precompiled chunks; precompiled chunks are not compatible between different Lua versions.

The standard paths in the official distribution may change between versions.

}

@sect2{@title{Incompatibilities in the Language}

@itemize{

@item{

The coercion of strings to numbers in arithmetic and bitwise operations has been removed from the core language.

The string library does a similar job for arithmetic (but not for bitwise) operations using the string metamethods.

However, unlike in previous versions, the new implementation preserves the implicit type of the numeral in the string.

For instance, the result of `@T{"1" + "2"}` now is an integer,  
not a float.  
}

@item{  
Literal decimal integer constants that overflow are read as floats,  
instead of wrapping around.  
You can use hexadecimal notation for such constants if you  
want  
the old behavior  
(reading them as integers with wrap around).  
}

@item{  
The use of the `@idx{__lt}` metamethod to emulate `@idx{__le}`  
has been removed.  
When needed, this metamethod must be explicitly defined.  
}

@item{  
The semantics of the numerical `@Rw{for}` loop  
over integers changed in some details.  
In particular, the control variable never wraps around.  
}

@item{  
A label for a `@Rw{goto}` cannot be declared where a label with the same  
name is visible, even if this other label is declared in an enclosing  
block.  
}

@item{  
When finalizing an object,  
Lua does not ignore `@idx{__gc}` metamethods that are not functions.  
Any value will be called, if present.  
(Non-callable values will generate a warning,  
like any other error when calling a finalizer.)  
}

}

}

@sect2{@title{Incompatibilities in the Libraries}  
@itemize{

@item{  
The function `@Lid{print}` does not call `@Lid{tostring}`

to format its arguments;  
instead, it has this functionality hardwired.  
You should use `@idx{__tostring}` to modify how values  
are printed.  
}

@item{  
The pseudo-random number generator used by the function `@Lid{math.random}`  
now starts with a somewhat random seed.  
Moreover, it uses a different algorithm.  
}

@item{  
By default, the decoding functions in the `@Lid{utf8}` library  
do not accept surrogates as valid code points.  
An extra parameter in these functions makes them more permissive.  
}

@item{  
The options `@St{setpause}` and `@St{setstepmul}`  
of the function `@Lid{collectgarbage}` are deprecated.  
You should use the new option `@St{incremental}` to set them.  
}

@item{  
The function `@Lid{io.lines}` now returns four values,  
instead of just one.  
That can be a problem when it is used as the sole  
argument to another function that has optional parameters,  
such as in `@T{load(io.lines(filename, "L"))}`.  
To fix that issue,  
you can wrap the call into parentheses,  
to adjust its number of results to one.  
}

}

}

@sect2{@title{Incompatibilities in the API}

@itemize{

@item{  
Full userdata now has an arbitrary number of associated  
user values.  
Therefore, the functions `@id{lua_newuserdata}`,  
`@id{lua_setuservalue}`, and `@id{lua_getuservalue}` were

replaced by @Lid{lua\_newuserdatauv},  
@Lid{lua\_setiuservalue}, and @Lid{lua\_getiuservalue},  
which have an extra argument.

For compatibility, the old names still work as macros assuming  
one single user value.

Note, however, that userdata with zero user values  
are more efficient memory-wise.

}

@item{

The function @Lid{lua\_resume} has an extra parameter.

This out parameter returns the number of values on  
the top of the stack that were yielded or returned by the coroutine.

(In previous versions,  
those values were the entire stack.)

}

@item{

The function @Lid{lua\_version} returns the version number,  
instead of an address of the version number.

The Lua core should work correctly with libraries using their  
own static copies of the same core,  
so there is no need to check whether they are using the same  
address space.

}

@item{

The constant @id{LUA\_ERRGCMM} was removed.

Errors in

finalizers are never propagated;  
instead, they generate a warning.

}

@item{

The options @idx{LUA\_GCSETPAUSE} and @idx{LUA\_GCSETSTEPMUL}  
of the function @Lid{lua\_gc} are deprecated.

You should use the new option @id{LUA\_GCINC} to set them.

}

}

}

}

@C{[=====]}

@sect1{BNF| @title{The Complete Syntax of Lua}

Here is the complete syntax of Lua in extended BNF.

As usual in extended BNF,

@bnfNter{A} means 0 or more @bnfNter{A}s,

and @bnfNter{[A]} means an optional @bnfNter{A}.

(For operator precedences, see @See{prec};

for a description of the terminals

@bnfNter{Name}, @bnfNter{Numeral},

and @bnfNter{LiteralString}, see @See{lexical}.)

@index{grammar}

@Produc{

@producname{chunk}@producbody{block}

@producname{block}@producbody{@bnfrep{stat} @bnfopt{retstat}}

@producname{stat}@producbody{

@bnfter{;}

@OrNL varlist @bnfter{=} explist

@OrNL functioncall

@OrNL label

@OrNL @Rw{break}

@OrNL @Rw{goto} Name

@OrNL @Rw{do} block

@Rw{end}

@OrNL @Rw{while} exp @Rw{do} block @Rw{end}

@OrNL @Rw{repeat} block @Rw{until} exp

@OrNL @Rw{if} exp @Rw{then} block

@bnfrep{@Rw{elseif} exp @Rw{then} block}

@bnfopt{@Rw{else} block} @Rw{end}

@OrNL @Rw{for} @bnfNter{Name} @bnfter{=} exp @bnfter{,} exp @bnfopt{@bnfter{,} exp}

@Rw{do} block @Rw{end}

@OrNL @Rw{for} namelist @Rw{in} explist @Rw{do} block @Rw{end}

@OrNL @Rw{function} funcname funcbody

@OrNL @Rw{local} @Rw{function} @bnfNter{Name} funcbody

@OrNL @Rw{local} attnamelist @bnfopt{@bnfter{=} explist}

}

@producname{attnamelist}@producbody{

@bnfNter{Name} attrib @bnfrep{@bnfter{,} @bnfNter{Name} attrib}}

@producname{attrib}@producbody{@bnfopt{@bnfter{<} @bnfNter{Name} @bnfter{>}}}

@producname{retstat}@producbody{@Rw{return}

@bnfopt{explist} @bnfopt{@bnfter{;}}}

@producname{label}@producbody{@bnfter{::} Name @bnfter{::}}

@producname{funcname}@producbody{@bnfNter{Name} @bnfrep{@bnfter{.} @bnfNter{Name}}}

@bnfopt{@bnfter{:} @bnfNter{Name}}}

@producname{varlist}@producbody{var @bnfrep{@bnfter{,} var}}

@producname{var}@producbody{  
@bnfNter{Name}  
@Or prefixexp @bnfter{[] exp @bnfter{[]}  
@Or prefixexp @bnfter{.} @bnfNter{Name}  
}

@producname{namelist}@producbody{@bnfNter{Name} @bnfrep{@bnfter{,} @bnfNter{Name}}}

@producname{explist}@producbody{exp @bnfrep{@bnfter{,} exp}}

@producname{exp}@producbody{  
@Rw{nil}  
@Or @Rw{false}  
@Or @Rw{true}  
@Or @bnfNter{Numeral}  
@Or @bnfNter{LiteralString}  
@Or @bnfter{...}  
@Or functiondef  
@OrNL prefixexp  
@Or tableconstructor  
@Or exp binop exp  
@Or unop exp  
}

@producname{prefixexp}@producbody{var @Or functioncall @Or @bnfter{() exp @bnfter{}}}

@producname{functioncall}@producbody{  
prefixexp args  
@Or prefixexp @bnfter{:} @bnfNter{Name} args  
}

@producname{args}@producbody{  
@bnfter{() @bnfopt{explist} @bnfter{}}  
@Or tableconstructor  
@Or @bnfNter{LiteralString}  
}

@producname{functiondef}@producbody{@Rw{function} funcbody}

```

@producname{funcbody}@producbody{@bnfter{()
@bnfopt{parlist} @bnfter{)}} block @Rw{end}}

@producname{parlist}@producbody{namelist @bnfopt{@bnfter{,} @bnfter{...}}
@Or @bnfter{...}}

@producname{tableconstructor}@producbody{@bnfter{@Open} @bnfopt{fieldlist} @bnfter{@Close}}

@producname{fieldlist}@producbody{field @bnfrep{fieldsep field} @bnfopt{fieldsep}}

@producname{field}@producbody{@bnfter{[] exp @bnfter{[] @bnfter{=} exp @Or @bnfNter{Name}
@bnfter{=} exp @Or exp}

@producname{fieldsep}@producbody{@bnfter{,} @Or @bnfter{;}}

@producname{binop}@producbody{
@bnfter{+} @Or @bnfter{-} @Or @bnfter{*} @Or @bnfter{/} @Or @bnfter{//}
@Or @bnfter{^} @Or @bnfter{%}
@OrNL
@bnfter{&} @Or @bnfter{~} @Or @bnfter{[] @Or @bnfter{>>} @Or @bnfter{<<}
@Or @bnfter{..}
@OrNL
@bnfter{<} @Or @bnfter{<=} @Or @bnfter{>} @Or @bnfter{>=}
@Or @bnfter{==} @Or @bnfter{~=}
@OrNL
@Rw{and} @Or @Rw{or}}

@producname{unop}@producbody{@bnfter{-} @Or @Rw{not} @Or @bnfter{#} @Or
@bnfter{~}}

}

}

@C{}}=====}

}
@C{}}-----}

```

Found

in path(s):

\* /lua-5-4-7-1-tar-gz/lua-5.4.7/manual/manual.of

No license file was found, but licenses were detected in source scan.

/\*\*\*\*\*

\* Copyright (C) 1994-2024 Lua.org, PUC-Rio.

\*

\* Permission is hereby granted, free of charge, to any person obtaining  
 \* a copy of this software and associated documentation files (the  
 \* "Software"), to deal in the Software without restriction, including  
 \* without limitation the rights to use, copy, modify, merge, publish,  
 \* distribute, sublicense, and/or sell copies of the Software, and to  
 \* permit persons to whom the Software is furnished to do so, subject to  
 \* the following conditions:  
 \*  
 \* The above copyright notice and this permission notice shall be  
 \* included in all copies or substantial portions of the Software.  
 \*  
 \* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,  
 \* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF  
 \* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.  
 \* IN NO EVENT  
 \* SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY  
 \* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,  
 \* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE  
 \* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

\*\*\*\*\*/

Found in path(s):

\* /lua-5-4-7-1-tar-gz/lua-5.4.7/lua.h

No license file was found, but licenses were detected in source scan.

--[[

\*\*\*\*\*

\* Copyright (C) 1994-2016 Lua.org, PUC-Rio.

\*

\* Permission is hereby granted, free of charge, to any person obtaining  
 \* a copy of this software and associated documentation files (the  
 \* "Software"), to deal in the Software without restriction, including  
 \* without limitation the rights to use, copy, modify, merge, publish,  
 \* distribute, sublicense, and/or sell copies of the Software, and to  
 \* permit persons to whom the Software is furnished to do so, subject to  
 \* the following conditions:

\*

\* The above copyright notice and this permission notice shall be  
 \* included in all copies or substantial portions of the Software.

\*

\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,  
 \* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF  
 \* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.  
 \* IN NO EVENT

\* SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY  
 \* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,  
 \* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE  
 \* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



\*\*\*\*\*

]]

Found in path(s):

\* /lua-5-4-7-1-tar-gz/lua-5.4.7/testes/all.lua

No license file was found, but licenses were detected in source scan.

&copy; 2024 Lua.org, PUC-Rio. All rights reserved.

header = string.gsub(header, "All rights reserved",

Found in path(s):

\* /lua-5-4-7-1-tar-gz/lua-5.4.7/manual/2html

# 1.130 utmps 0.1.3.1-r0

## 1.130.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you

distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program).

Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim  
copies of the Program's

source code as you receive it, in any medium, provided that you  
conspicuously and appropriately publish on each copy an appropriate  
copyright notice and disclaimer of warranty; keep intact all the  
notices that refer to this License and to the absence of any warranty;  
and give any other recipients of the Program a copy of this License  
along with the Program.

You may charge a fee for the physical act of transferring a copy, and  
you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion  
of it, thus forming a work based on the Program, and copy and  
distribute such modifications or work under the terms of Section 1  
above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices  
stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in

whole or in part contains or is derived from the Program or any  
part thereof, to be licensed as a whole at no charge to all third  
parties under the terms of this License.

c) If the modified program normally reads commands interactively  
when run, you must cause it, when started running for such  
interactive use in the most ordinary way, to print or display an  
announcement including an appropriate copyright notice and a  
notice that there is no warranty (or else, saying that you provide  
a warranty) and that users may redistribute the program under  
these conditions, and telling the user how to view a copy of this  
License. (Exception: if the Program itself is interactive but  
does not normally print such an announcement, your work based on  
the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If  
identifiable sections of that work are not derived from the Program,  
and can be reasonably

considered independent and separate works in  
themselves, then this License, and its terms, do not apply to those  
sections when you distribute them as separate works. But when you  
distribute the same sections as part of a whole which is a work based  
on the Program, the distribution of the whole must be on the terms of  
this License, whose permissions for other licensees extend to the

entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent

access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6.

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other

circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

GNU General Public License for more details.

You should have received a copy of the GNU General Public License  
along with this program; if not, write to the Free Software  
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this  
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.

The hypothetical commands

`show w' and `show c' should show the appropriate  
parts of the General Public License. Of course, the commands you use may  
be called something other than `show w' and `show c'; they could even be  
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your  
school, if any, to sign a "copyright disclaimer" for the program, if  
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into  
proprietary programs. If your program is a subroutine library, you may  
consider it more useful to permit linking proprietary applications with the  
library. If this is what you want to do, use the GNU Library General  
Public License instead of this License.

```
# Automatically generated by apkgbuild-cpan, template 4
# Contributor: Valery Kartel <valery.kartel@gmail.com>
# Maintainer: Celeste <cielesti@protonmail.com>
maintainer="Celeste <cielesti@protonmail.com>"
pkgname=perl-bsd-resource
pkgver=1.2911
pkgrel=10
#_pkgreal is used by apkgbuild-cpan to find modules at MetaCpan
_pkgreal=BSD-Resource
pkgdesc="Perl extension implements the BSD process resource limit functions"
```



```

url="https://metacpan.org/release/BSD-Resource/"
arch="all"
license="Artistic-2.0 OR LGPL-2.0"
depends="perl"
makedepends="perl-dev"
subpackages="$pkgname-doc"
source="https://cpan.metacpan.org/authors/id/J/JH/JHI/BSD-Resource-$pkgver.tar.gz"
builddir="$srcdir/$_pkgreal-$pkgver"

build() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    PERL_MM_USE_DEFAULT=1 perl -I. Makefile.PL \
        INSTALLDIRS=vendor \
        NO_PACKLIST=1 \
        NO_PERLLOCAL=1
    make
}

check() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    make test
}

package() {
    make DESTDIR="$pkgdir" install
}

sha512sums="
d0032d41c7c0468ed1c6d8f57b885f6cb97a5039d754c8cb60b2067daedaf53bd15fb6561a3d0f828df16dfa5417f663b
8065ba65f4fea16dc9262728b3b6b85
    BSD-Resource-1.2911.tar.gz
"

# Contributor: Maxim Karasev <begs@disroot.org>
# Maintainer: mio <miyopan@e.email>
pkgname=bsd-games
pkgver=3.3
pkgrel=2
pkgdesc="traditional text mode games from BSD"
url="https://bsd-games.sourceforge.io/"
arch="all"
license="BSD-3-Clause"
install="$pkgname.post-install"
makedepends="ncurses-dev coreutils" # configure causes busybox expr error
subpackages="$pkgname-doc"
source="https://sourceforge.net/projects/bsd-games/files/bsd-games-$pkgver.tar.gz
0001-bsd-games-scores-dir.patch
"

options="!check" # no tests

```

```

build() {
# some GNU autoconf options are ignored, but it works
./configure \
--build=$CBUILD \
--host=$CHOST \
--prefix=/usr \
--sysconfdir=/etc \
--mandir=/usr/share/man \
--sourcedir=/usr/share/$pkgname
make
}

package() {
make DESTDIR="$pkgdir" install
install -Dm644 LICENSE "$pkgdir"/usr/share/licenses/bsd-games/LICENSE
}

```

```

sha512sums="
aaf36d09d4fe68514a5c279063d7e77a9a9a84c447037cba974b29faa07eb80c2aab2379d0699c196b5c27ffe1b2bb2c6
8d3390143e0e62d5e3d210ccfb61294
bsd-games-3.3.tar.gz
e1def0d4835a877242c4173fae862e278a7858738e0bacacbb045a42bcabec2aa7c72d000bbd2421ce35fe66cdc1f52c0
577b7e278029ca5ed50f7aab22319f6 0001-bsd-games-scores-dir.patch
"

```

MIT License

Copyright (c) 2023 Leaf Corcoran

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```

# Maintainer: Natanael Copa <ncopa@alpinelinux.org>
pkgname=libart-lgpl
pkgver=2.3.21
pkgrel=8
pkgdesc="A library for high-performance 2D graphics"
url="https://www.levien.com/libart/"
arch="all"
options="!check" # No test suite.
license="LGPL-2.0-or-later"
subpackages="$pkgname-dev"
source="https://download.gnome.org/sources/libart-lgpl/2.3/libart-lgpl-$pkgver.tar.bz2"
"

builddir="$srcdir"/libart-lgpl-$pkgver

prepare() {
    update_config_sub
    default_prepare
}

build() {
    ./configure \
        --build=$CBUILD \
        --host=$CHOST \
        --prefix=/usr
    make
}

package() {
    make DESTDIR="$pkgdir" install
}

sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020 libart-lgpl-2.3.21.tar.bz2"
mini_sendmail - accept email on behalf of real sendmail

```

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.  
 All rights reserved.

Redistribution and use in source and binary forms, with or without  
 modification, are permitted provided that the following conditions  
 are met:

1. Redistributions of source code must retain the above copyright  
 notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright  
 notice, this list of conditions and the following disclaimer in the  
 documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# Contributor: Carlo Landmeter <clandmeter@alpinelinux.org>

# Maintainer: Kevin Daudt <kdaudt@alpinelinux.org>

pkgname=spdx-licenses

pkgver=3.26.0

pkgrel=0

pkgdesc="Various data formats for the SPDX License List"

url="https://spdx.org/"

arch="noarch"

options="!check" # no test suite

license="CC-BY-3.0"

source="license-list-data-\$pkgver.tar.gz::https://github.com/spdx/license-list-data/archive/v\$pkgver.tar.gz"

builddir="\$srcdir/license-list-data-\$pkgver"

subpackages="\$pkgname-list"

\_types="html json rdfa rdft rdfturtle rdfxml template text"

for type in \$\_types; do

subpackages="\$subpackages \$pkgname-\$type:\_subpkg"

done

package() {

mkdir -p "\$pkgdir"

}

\_subpkg() {

local type=\${subpkgname/\$pkgname-/}

pkgdesc="\$pkgdesc (\$type)"

install\_if="\$pkgname=\$pkgver-r\$pkgrel"

mkdir -p "\$subpkgdir"/usr/share/spdx

cp -r "\$builddir"/\$type "\$subpkgdir"/usr/share/spdx/

}

list() {

pkgdesc="\$pkgdesc (licence list)"

mkdir -p "\$subpkgdir"/usr/share/spdx

local i; for i in "\$builddir"/text/\*.txt;

```

do
  local license=${i##*/}
  echo ${license%.*} >> "$subpkgdir"/usr/share/spdx/license.lst
done
}

sha512sums="
67e618d4642dfe4f366935d9514d6f7941d3711e77c49bc1b340ac572d36015981b8f10c72a9f80f1d83e132ae7e5093c
0155e0c0e786e4777ee16d1cb7b885a  license-list-data-3.26.0.tar.gz
"

# Contributor: Patrycja Rosa <alpine@ptrcnll.me>
# Maintainer: Patrycja Rosa <alpine@ptrcnll.me>
pkgname=bsd-compat-headers
pkgver=0.7.2
pkgrel=6
pkgdesc="BSD compatibility headers (cdefs, queue, tree)"
url="https://gitlab.alpinelinux.org/alpine/aports"
arch="noarch"
license="BSD-2-Clause AND BSD-3-Clause"
source="
cdefs.h
queue.h
tree.h
"

builddir="$srcdir"
options="!check" # just headers

package() {
  mkdir -p "$pkgdir"
  install -Dm644 -t "$pkgdir"/usr/include/sys \
    cdefs.h queue.h tree.h
}

sha512sums="
37c8fc73c7aea7b490f7850927e2bb91d12137c9e59e22c084146d515696dbc7973b5de92f4c987ba080dd2502ba8312
7006442c3f019b6447a620c0cae73178  cdefs.h
2f0d5e6e4dc3350285cf17009265dddcbe12431c111868eea39bc8cb038ab7c1f2acacbb21735c4e9d4a1fd106a8fc0f86
11ea33987d4faba37dde5ce6da0750  queue.h
d9ac210d81feb8ad2655bc80fb065d3fe20ae4417b32b4a1711e6738a4870140005c13373b5d1846ef3ce5ae6da45f2da
cef2092881eded0a2e94f6a07752ef3  tree.h
"

Original Copyright (c) 1985 by Supoj Sutanthavibul
Parts Copyright (c) 1989-2012 by Brian V. Smith
Parts Copyright (c) 1991 by Paul King
Other Copyrights may be found in various files

```

Main Xfig copyright notice:

Any party obtaining a copy of these files is granted, free of charge, a full and unrestricted irrevocable, world-wide,

paid up, royalty-free, nonexclusive right and license to deal in this software and documentation files (the "Software"), including without limitation the rights to use, copy, modify, merge, publish and/or distribute copies of the Software, and to permit persons who receive copies from any such party to do so, with the only requirement being that this copyright notice remain intact.

No representations are made about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Parts Copyright (C) 1993 by Alan Richardson (mppa3@uk.ac.sussex.syma)

The text rotation code in w\_rottext.c was written by Alan Richardson.

The above copyright notice holds for this work as well.

Parts Copyright (c) 1994 by Anthony Dekker

The Kohonen neural network code for color optimization was written by Anthony Dekker.

Parts Copyright (c) 1995 by C. Blanc and C. Schlick

The X-Spline code was written Carole Blanc (blanc@labri.u-bordeaux.fr) and Christophe Schlick (schlick@labri.u-bordeaux.fr) starting from an initial implementation done by C. Feuille, S. Grosbois, L. Maziere and L. Minihot as a student practice (Universite Bordeaux, France). For additional information about X-splines, see:

"X-Splines: A Spline Model Designed for the End User" by C. Blanc and C. Schlick, Proceedings of SIGGRAPH'95

<http://dept-info.labri.u-bordeaux.fr/~schlick/DOC/sig1.html>

# Contributor: Dhruvin Gandhi <contact@dhruvin>

# Maintainer: Patrycja Rosa <alpine@ptrcnull.me>

pkgname=py3-license-expression

\_pkgname=license\_expression

pkgver=30.4.1

pkgrel=0

pkgdesc="Library to parse, compare, simplify and normalize license expressions"

url="https://github.com/nexB/license-expression"

arch="noarch"

license="Apache-2.0"

depends="python3 py3-boolean.py"

makedepends="py3-setuptools py3-setuptools\_scm py3-gpep517 py3-wheel"

checkdepends="py3-pytest py3-pytest-xdist"

subpackages="\$pkgname-pyc"

source="https://files.pythonhosted.org/packages/source/l/\_pkgname/\_pkgname-\$pkgver.tar.gz"

builddir="\$srcdir/\_pkgname-\$pkgver"

build() {

gpep517 build-wheel \

--wheel-dir .dist \

--output-fd 3 3>&1 >&2

}

check() {

python3 -m venv --clear --without-pip --system-site-packages .testenv

.testenv/bin/python3 -m installer .dist/\*.whl

.testenv/bin/python3 -m pytest

```

}

package() {
    python3 -m installer -d "$pkgdir" \
        .dist/*.whl
}

sha512sums="
114c630f098939757e49187e35b4e54883266a2ee9ca283724992420ef4c860f9606a8342da0ab74d93627c10805f312
fda3f699cbc7c4fbd47a71e220440fb0
    license_expression-30.4.1.tar.gz
"

/* $NetBSD: queue.h,v 1.70 2015/11/02 15:21:23 christos Exp $ */

/*
 * Copyright (c) 1991, 1993
 * The Regents of the University of California. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the University nor the names of its contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
 * BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * @(#)queue.h 8.5 (Berkeley) 8/20/94
 */

#ifndef _SYS_QUEUE_H_
#define _SYS_QUEUE_H_

```

```

/*
* This file defines five types of data structures: singly-linked lists,
* lists, simple queues, tail queues, and circular queues.
*
* A singly-linked list is headed by a single forward pointer. The
* elements are singly linked for minimum space and
* pointer manipulation
* overhead at the expense of O(n) removal for arbitrary elements. New
* elements can be added to the list after an existing element or at the
* head of the list. Elements being removed from the head of the list
* should use the explicit macro for this purpose for optimum
* efficiency. A singly-linked list may only be traversed in the forward
* direction. Singly-linked lists are ideal for applications with large
* datasets and few or no removals or for implementing a LIFO queue.
*
* A list is headed by a single forward pointer (or an array of forward
* pointers for a hash table header). The elements are doubly linked
* so that an arbitrary element can be removed without a need to
* traverse the list. New elements can be added to the list before
* or after an existing element or at the head of the list. A list
* may only be traversed in the forward direction.
*
* A simple queue is headed by a pair of pointers, one the head of the
* list and the
* other to the tail of the list. The elements are singly
* linked to save space, so elements can only be removed from the
* head of the list. New elements can be added to the list after
* an existing element, at the head of the list, or at the end of the
* list. A simple queue may only be traversed in the forward direction.
*
* A tail queue is headed by a pair of pointers, one to the head of the
* list and the other to the tail of the list. The elements are doubly
* linked so that an arbitrary element can be removed without a need to
* traverse the list. New elements can be added to the list before or
* after an existing element, at the head of the list, or at the end of
* the list. A tail queue may be traversed in either direction.
*
* A circle queue is headed by a pair of pointers, one to the head of the
* list and the other to the tail of the list. The elements are doubly
* linked so that an arbitrary element can be removed without a need to
* traverse the list. New elements
* can be added to the list before or after
* an existing element, at the head of the list, or at the end of the list.
* A circle queue may be traversed in either direction, but has a more
* complex end of list detection.
*
* For details on the use of these macros, see the queue(3) manual page.

```



```

*/

/*
* Include the definition of NULL only on NetBSD because sys/null.h
* is not available elsewhere. This conditional makes the header
* portable and it can simply be dropped verbatim into any system.
* The caveat is that on other systems some other header
* must provide NULL before the macros can be used.
*/
#ifdef __NetBSD__
#include <sys/null.h>
#endif

#if defined(QUEUEDEBUG)
# if defined(_KERNEL)
#  define QUEUEDEBUG_ABORT(...) panic(__VA_ARGS__)
# else
#  include <err.h>
#  define QUEUEDEBUG_ABORT(...) err(1, __VA_ARGS__)
# endif
#endif

/*
* Singly-linked List definitions.
*/
#define SLIST_HEAD(name, type) \
struct name { \
    struct type *slh_first; /* first \
    element */ \
}

#define SLIST_HEAD_INITIALIZER(head) \
{ NULL }

#define SLIST_ENTRY(type) \
struct { \
    struct type *sle_next; /* next element */ \
}

/*
* Singly-linked List access methods.
*/
#define SLIST_FIRST(head) ((head)->slh_first)
#define SLIST_END(head) NULL
#define SLIST_EMPTY(head) ((head)->slh_first == NULL)
#define SLIST_NEXT(elm, field) ((elm)->field.sle_next)

#define SLIST_FOREACH(var, head, field) \

```

```

for((var) = (head)->slh_first;  \
    (var) != SLIST_END(head);  \
    (var) = (var)->field.sle_next)

#define SLIST_FOREACH_SAFE(var, head, field, tvar) \
for ((var) = SLIST_FIRST((head));  \
    (var) != SLIST_END(head) &&  \
    ((tvar) = SLIST_NEXT((var), field), 1);  \
    (var) = (tvar))

/*
 * Singly-linked List functions.
 */
#define SLIST_INIT(head) do {  \
    (head)->slh_first = SLIST_END(head);  \
} while (/*CONSTCOND*/0)

#define SLIST_INSERT_AFTER(slistelm, elm, field) do {  \
    (elm)->field.sle_next = (slistelm)->field.sle_next;  \
    (slistelm)->field.sle_next  \
    = (elm);  \
} while (/*CONSTCOND*/0)

#define SLIST_INSERT_HEAD(head, elm, field) do {  \
    (elm)->field.sle_next = (head)->slh_first;  \
    (head)->slh_first = (elm);  \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE_AFTER(slistelm, field) do {  \
    (slistelm)->field.sle_next =  \
        SLIST_NEXT(SLIST_NEXT((slistelm), field), field);  \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE_HEAD(head, field) do {  \
    (head)->slh_first = (head)->slh_first->field.sle_next;  \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE(head, elm, type, field) do {  \
    if ((head)->slh_first == (elm)) {  \
        SLIST_REMOVE_HEAD((head), field);  \
    }  \
    else {  \
        struct type *curelm = (head)->slh_first;  \
        while(curelm->field.sle_next != (elm))  \
            curelm = curelm->field.sle_next;  \
        curelm->field.sle_next =  \
            curelm->field.sle_next->field.sle_next;  \
    }  \
}

```

```

} while (/*CONSTCOND*/0)

/*
 * List definitions.
 */
#define LIST_HEAD(name, type) \
struct name { \
    struct type *lh_first; /* first element */ \
}

#define LIST_HEAD_INITIALIZER(head) \
{ NULL }

#define LIST_ENTRY(type) \
struct { \
    struct type *le_next; /* next element */ \
    struct type **le_prev; /* address of previous next element */ \
}

/*
 * List access methods.
 */
#define LIST_FIRST(head) ((head)->lh_first)
#define LIST_END(head) NULL
#define LIST_EMPTY(head) ((head)->lh_first == LIST_END(head))
#define LIST_NEXT(elm, field) ((elm)->field.le_next)

#define LIST_FOREACH(var, head, field) \
for ((var) = ((head)->lh_first); \
     (var) != LIST_END(head); \
     (var) = ((var)->field.le_next))

#define LIST_FOREACH_SAFE(var, head, field, tvar) \
for ((var) = LIST_FIRST((head)); \
     (var) != LIST_END(head) && \
     ((tvar) = LIST_NEXT((var), field), 1); \
     (var) = (tvar))

#define LIST_MOVE(head1, head2) do { \
    LIST_INIT((head2)); \
    if (!LIST_EMPTY((head1))) \
    { \
        (head2)->lh_first = (head1)->lh_first; \
        LIST_INIT((head1)); \
    } \
} while (/*CONSTCOND*/0)

```

```

/*
 * List functions.
 */
#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_LIST_INSERT_HEAD(head, elm, field) \
    if ((head)->lh_first && \
        (head)->lh_first->field.le_prev != &(head)->lh_first) \
        QUEUEDEBUG_ABORT("LIST_INSERT_HEAD %p %s:%d", (head), \
            __FILE__, __LINE__);
#define QUEUEDEBUG_LIST_OP(elm, field) \
    if ((elm)->field.le_next && \
        (elm)->field.le_next->field.le_prev != \
            &(elm)->field.le_next) \
        QUEUEDEBUG_ABORT("LIST_* forw %p %s:%d", (elm), \
            __FILE__, __LINE__); \
    if (*(elm)->field.le_prev != (elm)) \
        QUEUEDEBUG_ABORT("LIST_* back %p %s:%d", (elm), \
            __FILE__, __LINE__);
#define QUEUEDEBUG_LIST_POSTREMOVE(elm, field) \
    (elm)->field.le_next = (void *)1L; \
    (elm)->field.le_prev = (void *)1L;
#else
#define QUEUEDEBUG_LIST_INSERT_HEAD(head, elm, field)
#define QUEUEDEBUG_LIST_OP(elm,
    field)
#define QUEUEDEBUG_LIST_POSTREMOVE(elm, field)
#endif

#define LIST_INIT(head) do { \
    (head)->lh_first = LIST_END(head); \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_AFTER(listelm, elm, field) do { \
    QUEUEDEBUG_LIST_OP((listelm), field) \
    if (((elm)->field.le_next = (listelm)->field.le_next) != \
        LIST_END(head)) \
        (listelm)->field.le_next->field.le_prev = \
            &(elm)->field.le_next; \
    (listelm)->field.le_next = (elm); \
    (elm)->field.le_prev = &(listelm)->field.le_next; \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_BEFORE(listelm, elm, field) do { \
    QUEUEDEBUG_LIST_OP((listelm), field) \
    (elm)->field.le_prev = (listelm)->field.le_prev; \
    (elm)->field.le_next = (listelm); \
    *(listelm)->field.le_prev = (elm); \
    (listelm)->field.le_prev = &(elm)->field.le_next; \
} while (/*CONSTCOND*/0)

```

```

} while (/*CONSTCOND*/0)

#define LIST_INSERT_HEAD(head, elm, field) do { \
    QUEUEDEBUG_LIST_INSERT_HEAD((head), (elm), field) \
    if (((elm)->field.le_next \
    = (head)->lh_first) != LIST_END(head))\
        (head)->lh_first->field.le_prev = &(elm)->field.le_next;\
    (head)->lh_first = (elm); \
    (elm)->field.le_prev = &(head)->lh_first; \
} while (/*CONSTCOND*/0)

#define LIST_REMOVE(elm, field) do { \
    QUEUEDEBUG_LIST_OP((elm), field) \
    if ((elm)->field.le_next != NULL) \
        (elm)->field.le_next->field.le_prev = \
            (elm)->field.le_prev; \
    *(elm)->field.le_prev = (elm)->field.le_next; \
    QUEUEDEBUG_LIST_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

#define LIST_REPLACE(elm, elm2, field) do { \
    if (((elm2)->field.le_next = (elm)->field.le_next) != NULL) \
        (elm2)->field.le_next->field.le_prev = \
            &(elm2)->field.le_next; \
    (elm2)->field.le_prev = (elm)->field.le_prev; \
    *(elm2)->field.le_prev = (elm2); \
    QUEUEDEBUG_LIST_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

/*
 * Simple queue definitions.
 */
#define SIMPLEQ_HEAD(name, type) \
struct name { \
    struct \
    type *sqh_first; /* first element */ \
    struct type **sqh_last; /* addr of last next element */ \
}

#define SIMPLEQ_HEAD_INITIALIZER(head) \
{ NULL, &(head).sqh_first }

#define SIMPLEQ_ENTRY(type) \
struct { \
    struct type *sqe_next; /* next element */ \
}

/*

```

```

* Simple queue access methods.
*/

#define SIMPLEQ_FIRST(head) ((head)->sqh_first)
#define SIMPLEQ_END(head) NULL
#define SIMPLEQ_EMPTY(head) ((head)->sqh_first == SIMPLEQ_END(head))
#define SIMPLEQ_NEXT(elm, field) ((elm)->field.sqe_next)

#define SIMPLEQ_FOREACH(var, head, field) \
for ((var) = ((head)->sqh_first); \
     (var) != SIMPLEQ_END(head); \
     (var) = ((var)->field.sqe_next))

#define SIMPLEQ_FOREACH_SAFE(var, head, field, next) \
for ((var) = ((head)->sqh_first); \
     (var) != SIMPLEQ_END(head) && \
     ((next = ((var)->field.sqe_next)), 1); \
     (var) = (next))

/*
* Simple queue functions.
*/

#define SIMPLEQ_INIT(head) do { \
(head)->sqh_first = NULL; \
(head)->sqh_last \
= &(head)->sqh_first; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_HEAD(head, elm, field) do { \
if (((elm)->field.sqe_next = (head)->sqh_first) == NULL) \
(head)->sqh_last = &(elm)->field.sqe_next; \
(head)->sqh_first = (elm); \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_TAIL(head, elm, field) do { \
(elm)->field.sqe_next = NULL; \
*(head)->sqh_last = (elm); \
(head)->sqh_last = &(elm)->field.sqe_next; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_AFTER(head, listelm, elm, field) do { \
if (((elm)->field.sqe_next = (listelm)->field.sqe_next) == NULL) \
(head)->sqh_last = &(elm)->field.sqe_next; \
(listelm)->field.sqe_next = (elm); \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE_HEAD(head, field) do { \
if (((head)->sqh_first = (head)->sqh_first->field.sqe_next) == NULL) \
(head)->sqh_last = &(head)->sqh_first; \

```

```

} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE_AFTER(head, elm, field) do { \
    if \
    (((elm)->field.sqe_next = (elm)->field.sqe_next->field.sqe_next) \
     == NULL) \
    (head)->sqh_last = &(elm)->field.sqe_next; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE(head, elm, type, field) do { \
    if ((head)->sqh_first == (elm)) { \
        SIMPLEQ_REMOVE_HEAD((head), field); \
    } else { \
        struct type *curelm = (head)->sqh_first; \
        while (curelm->field.sqe_next != (elm)) \
            curelm = curelm->field.sqe_next; \
        if ((curelm->field.sqe_next = \
            curelm->field.sqe_next->field.sqe_next) == NULL) \
            (head)->sqh_last = &(curelm)->field.sqe_next; \
    } \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_CONCAT(head1, head2) do { \
    if (!SIMPLEQ_EMPTY((head2))) { \
        *(head1)->sqh_last = (head2)->sqh_first; \
        (head1)->sqh_last = (head2)->sqh_last; \
        SIMPLEQ_INIT((head2)); \
    } \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_LAST(head, type, field) \
    (SIMPLEQ_EMPTY((head)) ? \
     NULL : \
     ((struct type \
      *)((void *) \
        ((char *)((head)->sqh_last) - offsetof(struct type, field))))

/*
 * Tail queue definitions.
 */
#define _TAILQ_HEAD(name, type, qual) \
    struct name { \
        qual type *tqh_first; /* first element */ \
        qual type *qual *tqh_last; /* addr of last next element */ \
    }
#define TAILQ_HEAD(name, type) _TAILQ_HEAD(name, struct type,)

#define TAILQ_HEAD_INITIALIZER(head) \

```

```

{ TAILQ_END(head), &(head).tqh_first }

#define _TAILQ_ENTRY(type, qual) \
struct { \
    qual type *tqe_next; /* next element */ \
    qual type *qual *tqe_prev; /* address of previous next element */ \
}
#define TAILQ_ENTRY(type) _TAILQ_ENTRY(struct type,)

/*
 * Tail queue access methods.
 */
#define TAILQ_FIRST(head) ((head)->tqh_first)
#define TAILQ_END(head) (NULL)
#define TAILQ_NEXT(elm, field) ((elm)->field.tqe_next)
#define TAILQ_LAST(head, headname) \
    (((struct headname *) (void *) ((head)->tqh_last))->tqh_last)
#define TAILQ_PREV(elm, headname, field) \
    (((struct headname *) (void *) ((elm)->field.tqe_prev))->tqh_last)
#define TAILQ_EMPTY(head) (TAILQ_FIRST(head) == TAILQ_END(head))

#define TAILQ_FOREACH(var, head, field) \
    for ((var) = ((head)->tqh_first); \
        (var) != TAILQ_END(head); \
        (var) = ((var)->field.tqe_next))

#define TAILQ_FOREACH_SAFE(var, head, field, next) \
    for ((var) = ((head)->tqh_first); \
        (var) != TAILQ_END(head) && \
        ((next) = TAILQ_NEXT(var, field), 1); (var) = (next))

#define TAILQ_FOREACH_REVERSE(var, head, headname, field) \
    for ((var) = TAILQ_LAST((head), headname); \
        (var) != TAILQ_END(head); \
        (var) = TAILQ_PREV((var), headname, field))

#define TAILQ_FOREACH_REVERSE_SAFE(var, head, headname, field, prev) \
    for ((var) = TAILQ_LAST((head), headname); \
        (var) != TAILQ_END(head) && \
        ((prev) = TAILQ_PREV((var), headname, field), 1); (var) = (prev))

/*
 * Tail queue functions.
 */
#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_TAILQ_INSERT_HEAD(head,

```



```

elm, field) \
if ((head)->tqh_first && \
    (head)->tqh_first->field.tqe_prev != &(head)->tqh_first) \
    QUEUEDEBUG_ABORT("TAILQ_INSERT_HEAD %p %s:%d", (head), \
        __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_INSERT_TAIL(head, elm, field) \
if (*(head)->tqh_last != NULL) \
    QUEUEDEBUG_ABORT("TAILQ_INSERT_TAIL %p %s:%d", (head), \
        __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_OP(elm, field) \
if ((elm)->field.tqe_next && \
    (elm)->field.tqe_next->field.tqe_prev != \
    &(elm)->field.tqe_next) \
    QUEUEDEBUG_ABORT("TAILQ_* forw %p %s:%d", (elm), \
        __FILE__, __LINE__); \
if (*(elm)->field.tqe_prev != (elm)) \
    QUEUEDEBUG_ABORT("TAILQ_* back %p %s:%d", (elm), \
        __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_PREREMOVE(head, elm, field) \
if ((elm)->field.tqe_next == NULL && \
    (head)->tqh_last != &(elm)->field.tqe_next) \
    QUEUEDEBUG_ABORT("TAILQ_PREREMOVE head %p elm %p %s:%d", \
        (head), (elm), __FILE__, \
        __LINE__);
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field) \
(elm)->field.tqe_next = (void *)1L; \
(elm)->field.tqe_prev = (void *)1L;
#else
#define QUEUEDEBUG_TAILQ_INSERT_HEAD(head, elm, field)
#define QUEUEDEBUG_TAILQ_INSERT_TAIL(head, elm, field)
#define QUEUEDEBUG_TAILQ_OP(elm, field)
#define QUEUEDEBUG_TAILQ_PREREMOVE(head, elm, field)
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field)
#endif

#define TAILQ_INIT(head) do { \
    (head)->tqh_first = TAILQ_END(head); \
    (head)->tqh_last = &(head)->tqh_first; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_HEAD(head, elm, field) do { \
    QUEUEDEBUG_TAILQ_INSERT_HEAD((head), (elm), field) \
    if (((elm)->field.tqe_next = (head)->tqh_first) != TAILQ_END(head)) \
        (head)->tqh_first->field.tqe_prev = \
            &(elm)->field.tqe_next; \
    else \
        (head)->tqh_last = &(elm)->field.tqe_next; \
    (head)->tqh_first = (elm); \
}

```

```

(elm)->field.tqe_prev = &(head)->tqh_first; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_TAIL(head,
elm, field) do { \
    QUEUEDEBBUG_TAILQ_INSERT_TAIL((head), (elm), field) \
    (elm)->field.tqe_next = TAILQ_END(head); \
    (elm)->field.tqe_prev = (head)->tqh_last; \
    *(head)->tqh_last = (elm); \
    (head)->tqh_last = &(elm)->field.tqe_next; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_AFTER(head, listelm, elm, field) do { \
    QUEUEDEBBUG_TAILQ_OP((listelm), field) \
    if (((elm)->field.tqe_next = (listelm)->field.tqe_next) != \
        TAILQ_END(head)) \
        (elm)->field.tqe_next->field.tqe_prev = \
            &(elm)->field.tqe_next; \
    else \
        (head)->tqh_last = &(elm)->field.tqe_next; \
    (listelm)->field.tqe_next = (elm); \
    (elm)->field.tqe_prev = &(listelm)->field.tqe_next; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_BEFORE(listelm, elm, field) do { \
    QUEUEDEBBUG_TAILQ_OP((listelm), field) \
    (elm)->field.tqe_prev = (listelm)->field.tqe_prev; \
    (elm)->field.tqe_next = (listelm); \
    *(listelm)->field.tqe_prev = (elm); \
    (listelm)->field.tqe_prev \
    = &(elm)->field.tqe_next; \
} while (/*CONSTCOND*/0)

#define TAILQ_REMOVE(head, elm, field) do { \
    QUEUEDEBBUG_TAILQ_PREREMOVE((head), (elm), field) \
    QUEUEDEBBUG_TAILQ_OP((elm), field) \
    if (((elm)->field.tqe_next) != TAILQ_END(head)) \
        (elm)->field.tqe_next->field.tqe_prev = \
            (elm)->field.tqe_prev; \
    else \
        (head)->tqh_last = (elm)->field.tqe_prev; \
    *(elm)->field.tqe_prev = (elm)->field.tqe_next; \
    QUEUEDEBBUG_TAILQ_POSTREMOVE((elm), field); \
} while (/*CONSTCOND*/0)

#define TAILQ_REPLACE(head, elm, elm2, field) do { \
    if (((elm2)->field.tqe_next = (elm)->field.tqe_next) != \
        TAILQ_END(head)) \

```

```

        (elm2)->field.tqe_next->field.tqe_prev = \
            &(elm2)->field.tqe_next; \
    else \
        (head)->tqh_last = &(elm2)->field.tqe_next; \
        (elm2)->field.tqe_prev = (elm)->field.tqe_prev; \
        *(elm2)->field.tqe_prev = (elm2); \
    QUEUEDEBUG_TAILQ_POSTREMOVE((elm),
    field); \
} while (/*CONSTCOND*/0)

#define TAILQ_CONCAT(head1, head2, field) do { \
    if (!TAILQ_EMPTY(head2)) { \
        *(head1)->tqh_last = (head2)->tqh_first; \
        (head2)->tqh_first->field.tqe_prev = (head1)->tqh_last; \
        (head1)->tqh_last = (head2)->tqh_last; \
        TAILQ_INIT((head2)); \
    } \
} while (/*CONSTCOND*/0)

/*
 * Singly-linked Tail queue declarations.
 */
#define STAILQ_HEAD(name, type) \
struct name { \
    struct type *stqh_first; /* first element */ \
    struct type **stqh_last; /* addr of last next element */ \
}

#define STAILQ_HEAD_INITIALIZER(head) \
{ NULL, &(head).stqh_first }

#define STAILQ_ENTRY(type) \
struct { \
    struct type *stqe_next; /* next element */ \
}

/*
 * Singly-linked Tail queue access methods.
 */
#define STAILQ_FIRST(head) ((head)->stqh_first)
#define STAILQ_END(head) NULL
#define STAILQ_NEXT(elm, field) ((elm)->field.stqe_next)
#define STAILQ_EMPTY(head) (STAILQ_FIRST(head) \
    == STAILQ_END(head))

/*
 * Singly-linked Tail queue functions.
 */

```

```

#define STAILQ_INIT(head) do { \
    (head)->stqh_first = NULL; \
    (head)->stqh_last = &(head)->stqh_first; \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_HEAD(head, elm, field) do { \
    if (((elm)->field.stqe_next = (head)->stqh_first) == NULL) \
        (head)->stqh_last = &(elm)->field.stqe_next; \
    (head)->stqh_first = (elm); \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_TAIL(head, elm, field) do { \
    (elm)->field.stqe_next = NULL; \
    *(head)->stqh_last = (elm); \
    (head)->stqh_last = &(elm)->field.stqe_next; \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_AFTER(head, listelm, elm, field) do { \
    if (((elm)->field.stqe_next = (listelm)->field.stqe_next) == NULL) \
        (head)->stqh_last = &(elm)->field.stqe_next; \
    (listelm)->field.stqe_next = (elm); \
} while (/*CONSTCOND*/0)

#define STAILQ_REMOVE_HEAD(head, field) do { \
    if (((head)->stqh_first = (head)->stqh_first->field.stqe_next) \
        == NULL) \
        (head)->stqh_last = &(head)->stqh_first; \
} while (/*CONSTCOND*/0)

#define STAILQ_REMOVE(head, elm, type, field) do { \
    if ((head)->stqh_first == (elm)) { \
        STAILQ_REMOVE_HEAD((head), field); \
    } else { \
        struct type *curelm = (head)->stqh_first; \
        while (curelm->field.stqe_next != (elm)) \
            curelm = curelm->field.stqe_next; \
        if ((curelm->field.stqe_next = \
            curelm->field.stqe_next->field.stqe_next) == NULL) \
            (head)->stqh_last = &(curelm)->field.stqe_next; \
    } \
} while (/*CONSTCOND*/0)

#define STAILQ_FOREACH(var, head, field) \
    for ((var) = ((head)->stqh_first); \
        (var); \
        (var) = ((var)->field.stqe_next))

#define STAILQ_FOREACH_SAFE(var, head, field, tvar) \

```

```

for ((var) = STAILQ_FIRST((head)); \
    (var) && ((tvar) = STAILQ_NEXT((var), field), 1); \
    (var) = (tvar))

#define STAILQ_CONCAT(head1, head2) do { \
    if (!STAILQ_EMPTY((head2))) { \
        *(head1)->stqh_last \
        = (head2)->stqh_first; \
        (head1)->stqh_last = (head2)->stqh_last; \
        STAILQ_INIT((head2)); \
    } \
} while (/*CONSTCOND*/0)

#define STAILQ_LAST(head, type, field) \
(STAILQ_EMPTY((head)) ? \
    NULL : \
    ((struct type *) (void *) \
    ((char *) ((head)->stqh_last) - offsetof(struct type, field))))

#ifdef _KERNEL
/*
 * Circular queue definitions. Do not use. We still keep the macros
 * for compatibility but because of pointer aliasing issues their use
 * is discouraged!
 */

/*
 * __launder_type(): We use this ugly hack to work around the the compiler
 * noticing that two types may not alias each other and elide tests in code.
 * We hit this in the CIRCLEQ macros when comparing 'struct name *' and
 * 'struct type *' (see CIRCLEQ_HEAD()). Modern compilers (such as GCC
 * 4.8) declare these comparisons as always false, causing the code to
 * not run as designed.
 *
 * This hack is only to be used for comparisons and thus can be fully const.
 * Do not use for assignment.
 *
 * If we ever choose to change the ABI of the CIRCLEQ macros, we could fix
 * this by changing the head/tail sentinel values, but see the note above
 * this one.
 */
static __inline const void * __launder_type(const void *);
static __inline const void *
__launder_type(const void *__x)
{
    __asm __volatile("" : "+r" (__x));
    return __x;
}

```

```

}

#if defined(QEUEDEBUG)
#define QEUEDEBUG_CIRCLEQ_HEAD(head, field) \
if ((head)->cqh_first != CIRCLEQ_ENDC(head) && \
    (head)->cqh_first->field.cqe_prev != CIRCLEQ_ENDC(head)) \
    QEUEDEBUG_ABORT("CIRCLEQ head forw %p %s:%d", (head), \
        __FILE__, __LINE__); \
if ((head)->cqh_last != CIRCLEQ_ENDC(head) && \
    (head)->cqh_last->field.cqe_next != CIRCLEQ_ENDC(head)) \
    QEUEDEBUG_ABORT("CIRCLEQ head back %p %s:%d", (head), \
        __FILE__, __LINE__);
#define QEUEDEBUG_CIRCLEQ_ELM(head, elm, field) \
if ((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) { \
    if ((head)->cqh_last != (elm)) \
        QEUEDEBUG_ABORT("CIRCLEQ \
elm last %p %s:%d", \
            (elm), __FILE__, __LINE__); \
} else { \
    if ((elm)->field.cqe_next->field.cqe_prev != (elm)) \
        QEUEDEBUG_ABORT("CIRCLEQ elm forw %p %s:%d", \
            (elm), __FILE__, __LINE__); \
} \
if ((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) { \
    if ((head)->cqh_first != (elm)) \
        QEUEDEBUG_ABORT("CIRCLEQ elm first %p %s:%d", \
            (elm), __FILE__, __LINE__); \
} else { \
    if ((elm)->field.cqe_prev->field.cqe_next != (elm)) \
        QEUEDEBUG_ABORT("CIRCLEQ elm prev %p %s:%d", \
            (elm), __FILE__, __LINE__); \
}
#define QEUEDEBUG_CIRCLEQ_POSTREMOVE(elm, field) \
(elm)->field.cqe_next = (void *)1L; \
(elm)->field.cqe_prev = (void *)1L;
#else
#define QEUEDEBUG_CIRCLEQ_HEAD(head, field)
#define QEUEDEBUG_CIRCLEQ_ELM(head, elm, field)
#define QEUEDEBUG_CIRCLEQ_POSTREMOVE(elm, field)
#endif

#define CIRCLEQ_HEAD(name, type) \
struct name { \
    struct type *cqh_first; /* \
first element */ \
    struct type *cqh_last; /* last element */ \
}

```

```

#define CIRCLEQ_HEAD_INITIALIZER(head)  \
{ CIRCLEQ_END(&head), CIRCLEQ_END(&head) }

#define CIRCLEQ_ENTRY(type)  \
struct {  \
    struct type *cqe_next; /* next element */  \
    struct type *cqe_prev; /* previous element */  \
}

/*
 * Circular queue functions.
 */
#define CIRCLEQ_INIT(head) do {  \
    (head)->cqh_first = CIRCLEQ_END(head);  \
    (head)->cqh_last = CIRCLEQ_END(head);  \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_AFTER(head, listelm, elm, field) do {  \
    QUEUEDEBDEBUG_CIRCLEQ_HEAD((head), field)  \
    QUEUEDEBDEBUG_CIRCLEQ_ELM((head), (listelm), field)  \
    (elm)->field.cqe_next = (listelm)->field.cqe_next;  \
    (elm)->field.cqe_prev = (listelm);  \
    if ((listelm)->field.cqe_next == CIRCLEQ_ENDC(head))  \
        (head)->cqh_last = (elm);  \
    else  \
        (listelm)->field.cqe_next->field.cqe_prev = (elm);  \
    (listelm)->field.cqe_next = (elm);  \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_BEFORE(head, listelm, elm, field) do {  \
    QUEUEDEBDEBUG_CIRCLEQ_HEAD((head), field)  \
    QUEUEDEBDEBUG_CIRCLEQ_ELM((head), (listelm), field)  \
    (elm)->field.cqe_next = (listelm);  \
    (elm)->field.cqe_prev = (listelm)->field.cqe_prev;  \
    if ((listelm)->field.cqe_prev == CIRCLEQ_ENDC(head))  \
        (head)->cqh_first = (elm);  \
    else  \
        (listelm)->field.cqe_prev->field.cqe_next = (elm);  \
    (listelm)->field.cqe_prev = (elm);  \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_HEAD(head, elm, field) do {  \
    QUEUEDEBDEBUG_CIRCLEQ_HEAD((head), field)  \
    (elm)->field.cqe_next = (head)->cqh_first;  \
    (elm)->field.cqe_prev = CIRCLEQ_END(head);  \
    if ((head)->cqh_last == CIRCLEQ_ENDC(head))  \
        (head)->cqh_last = (elm);  \
} while (/*CONSTCOND*/0)

```

```

else    \
    (head)->cqh_first->field.cqe_prev = (elm); \
    (head)->cqh_first = (elm);    \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_TAIL(head, elm, field) do { \
    QUEUEDEBUG_CIRCLEQ_HEAD((head), field) \
    (elm)->field.cqe_next \
    = CIRCLEQ_END(head); \
    (elm)->field.cqe_prev = (head)->cqh_last; \
    if ((head)->cqh_first == CIRCLEQ_ENDC(head)) \
        (head)->cqh_first = (elm); \
    else \
        (head)->cqh_last->field.cqe_next = (elm); \
    (head)->cqh_last = (elm); \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_REMOVE(head, elm, field) do { \
    QUEUEDEBUG_CIRCLEQ_HEAD((head), field) \
    QUEUEDEBUG_CIRCLEQ_ELM((head), (elm), field) \
    if ((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) \
        (head)->cqh_last = (elm)->field.cqe_prev; \
    else \
        (elm)->field.cqe_next->field.cqe_prev = \
            (elm)->field.cqe_prev; \
    if ((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \
        (head)->cqh_first = (elm)->field.cqe_next; \
    else \
        (elm)->field.cqe_prev->field.cqe_next = \
            (elm)->field.cqe_next; \
    QUEUEDEBUG_CIRCLEQ_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_FOREACH(var, head, field) \
    for ((var) = ((head)->cqh_first); \
        (var) != CIRCLEQ_ENDC(head); \
        (var) \
        = ((var)->field.cqe_next))

#define CIRCLEQ_FOREACH_REVERSE(var, head, field) \
    for ((var) = ((head)->cqh_last); \
        (var) != CIRCLEQ_ENDC(head); \
        (var) = ((var)->field.cqe_prev))

/*
 * Circular queue access methods.
 */
#define CIRCLEQ_FIRST(head) ((head)->cqh_first)

```



```

#define CIRCLEQ_LAST(head) ((head)->cqh_last)
/* For comparisons */
#define CIRCLEQ_ENDC(head) (__lauder_type(head))
/* For assignments */
#define CIRCLEQ_END(head) ((void *)(head))
#define CIRCLEQ_NEXT(elm, field) ((elm)->field.cqe_next)
#define CIRCLEQ_PREV(elm, field) ((elm)->field.cqe_prev)
#define CIRCLEQ_EMPTY(head) \
    (CIRCLEQ_FIRST(head) == CIRCLEQ_ENDC(head))

#define CIRCLEQ_LOOP_NEXT(head, elm, field) \
    (((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) \
     ? ((head)->cqh_first) \
     : (elm->field.cqe_next))
#define CIRCLEQ_LOOP_PREV(head, elm, field) \
    (((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \
     ? ((head)->cqh_last) \
     : \
     (elm->field.cqe_prev))
#endif /* !_KERNEL */

#endif /* !_SYS_QUEUE_H_ */
# Contributor: Fabian Affolter <fabian@affolter-engineering.ch>
# Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>
pkgname=py3-flake8-copyright
_pkgname=flake8-copyright
pkgver=0.2.4
pkgrel=3
pkgdesc="Extension for flake8 which checks for copyrights"
options="!check" # No testsuite
url="https://github.com/savoirfairelinux/flake8-copyright"
arch="noarch"
license="MIT"
depends="py3-flake8 py3-setuptools"
makedepends="py3-gpep517 py3-wheel"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/${_pkgname:0:1}/${_pkgname}/${_pkgname}-${pkgver}.tar.gz"
buildid="$_srcdir"/$_pkgname-$pkgver

replaces="py-flake8-copyright" # Backwards compatibility
provides="py-flake8-copyright=$pkgver-r$pkgrel" # Backwards compatibility

build() {
    gpep517 build-wheel \
        --wheel-dir .dist \
        --output-fd 3 3>&1 >&2
}

```

```

package() {
    gpep517 install-wheel --destdir "$pkgdir" \
        .dist/*.whl
}

sha512sums="
a6ab47e1bb715618f075c51f398e18180404871b3b0faf1c2d30701d5203db6bc23771bf22ffb5bafcd8ee856b9b64237
b316bd503ae7dceed6ca284ccd5a74d
flake8-copyright-0.2.4.tar.gz
"

#ifndef _SYS_CDEFS_H_
#define _SYS_CDEFS_H_

#warning usage of non-standard #include <sys/cdefs.h> is deprecated

#undef __P
#undef __PMT

#define __P(args) args
#define __PMT(args) args

#define __CONCAT(x,y) x ## y
#define __STRING(x) #x

#ifdef __cplusplus
# define __BEGIN_DECLS extern "C" {
# define __END_DECLS }
#else
# define __BEGIN_DECLS
# define __END_DECLS
#endif

#if defined(__GNUC__) && !defined(__cplusplus)
# define __THROW __attribute__((__nothrow__))
# define __NTH(fct) __attribute__((__nothrow__)) fct
#else
# define __THROW
# define __NTH(fct) fct
#endif

#endif /* _SYS_CDEFS_H_ */

# Maintainer: fossdd <fossdd@pwned.life>
pkgname=cargo-license
pkgver=0.6.1
pkgrel=1
pkgdesc="Cargo subcommand to see license of dependencies"
url="https://github.com/onur/cargo-license"
arch="all"

```

```

license="MIT"
makedepends="cargo cargo-auditable"
source="$pkgname-$pkgver.tar.gz::https://crates.io/api/v1/crates/cargo-license/$pkgver/download"

prepare() {
    default_prepare

    cargo fetch --target="$CTARGET" --locked
}

build() {
    cargo auditable build --release --frozen
}

check() {
    cargo test --frozen
}

package() {
    install -Dm755 target/release/cargo-license -t "$pkgdir"/usr/bin
}

sha512sums="
36215145e79965be090ae2f5ce6f5eb208465eb67a94c26bed1fe70a131c46c6dc35695e30852759b7366883fbc3caf00
272d796ac634f0a5e1d49232b6b29d5  cargo-license-0.6.1.tar.gz
"

/* $NetBSD: tree.h,v 1.20 2013/09/14 13:20:45 joerg Exp $ */
/* $OpenBSD: tree.h,v 1.13 2011/07/09 00:19:45 pirofti Exp $ */
/* Modified by Void Linux. */
/*
* Copyright 2002 Niels Provos <provos@citi.umich.edu>
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE
* AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT

```

\* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,  
 \* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY  
 \* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT  
 \* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF  
 \* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.  
 \*/

```
#ifndef _SYS_TREE_H_
#define _SYS_TREE_H_
```

```
#ifdef __GNUC__
#define __GNUC_PREREQ__(x, y) \
  ((__GNUC__ == (x) && __GNUC_MINOR__ >= (y)) || \
   (__GNUC__ > (x)))
#else
#define __GNUC_PREREQ__(x, y) 0
#endif
```

```
#if __GNUC_PREREQ__(2, 7) || defined(__lint__)
#define _sys_tree_h_unused __attribute__((__unused__))
#else
#define _sys_tree_h_unused /* delete */
#endif
```

/\*

\* This file defines data structures for different types of trees:  
 \* splay trees and red-black trees.  
 \*  
 \* A splay tree is a self-organizing  
 data structure. Every operation  
 \* on the tree causes a splay to happen. The splay moves the requested  
 \* node to the root of the tree and partly rebalances it.  
 \*  
 \* This has the benefit that request locality causes faster lookups as  
 \* the requested nodes move to the top of the tree. On the other hand,  
 \* every lookup causes memory writes.  
 \*  
 \* The Balance Theorem bounds the total access time for m operations  
 \* and n inserts on an initially empty tree as  $O((m + n)\lg n)$ . The  
 \* amortized cost for a sequence of m accesses to a splay tree is  $O(\lg n)$ ;  
 \*  
 \* A red-black tree is a binary search tree with the node color as an  
 \* extra attribute. It fulfills a set of conditions:  
 \* - every search path from the root to a leaf consists of the  
 \* same number of black nodes,  
 \* - each red node (except for the root) has a black parent,  
 \* - each leaf node is black.  
 \*

```

* Every operation on a red-black tree is bounded as  $O(\lg n)$ .
* The maximum height of a red-black tree is  $2\lg(n+1)$ .
*/

#define SPLAY_HEAD(name, type) \
struct name { \
    struct type *sph_root; /* root of the tree */ \
}

#define SPLAY_INITIALIZER(root) \
{ NULL }

#define SPLAY_INIT(root) do { \
    (root)->sph_root = NULL; \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ENTRY(type) \
struct { \
    struct type *spe_left; /* left element */ \
    struct type *spe_right; /* right element */ \
}

#define SPLAY_LEFT(elm, field) (elm)->field.spe_left
#define SPLAY_RIGHT(elm, field) (elm)->field.spe_right
#define SPLAY_ROOT(head) (head)->sph_root
#define SPLAY_EMPTY(head) (SPLAY_ROOT(head) == NULL)

/* SPLAY_ROTATE_{LEFT,RIGHT} expect that tmp hold SPLAY_{RIGHT,LEFT} */
#define SPLAY_ROTATE_RIGHT(head, tmp, field) do { \
    SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(tmp, field); \
    SPLAY_RIGHT(tmp, field) = (head)->sph_root; \
    (head)->sph_root = tmp; \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ROTATE_LEFT(head, tmp, field) do { \
    SPLAY_RIGHT((head)->sph_root, \
    field) = SPLAY_LEFT(tmp, field); \
    SPLAY_LEFT(tmp, field) = (head)->sph_root; \
    (head)->sph_root = tmp; \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKLEFT(head, tmp, field) do { \
    SPLAY_LEFT(tmp, field) = (head)->sph_root; \
    tmp = (head)->sph_root; \
    (head)->sph_root = SPLAY_LEFT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKRIGHT(head, tmp, field) do { \

```

```

    SPLAY_RIGHT(tmp, field) = (head)->sph_root; \
    tmp = (head)->sph_root; \
    (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ASSEMBLE(head, node, left, right, field) do { \
    SPLAY_RIGHT(left, field) = SPLAY_LEFT((head)->sph_root, field); \
    SPLAY_LEFT(right, field) = SPLAY_RIGHT((head)->sph_root, field); \
    SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(node, field); \
    SPLAY_RIGHT((head)->sph_root, field) = SPLAY_LEFT(node, field); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */

#define
    SPLAY_PROTOTYPE(name, type, field, cmp) \
    void name##_SPLAY(struct name *, struct type *); \
    void name##_SPLAY_MINMAX(struct name *, int); \
    struct type *name##_SPLAY_INSERT(struct name *, struct type *); \
    struct type *name##_SPLAY_REMOVE(struct name *, struct type *); \
    \
/* Finds the node with the same key as elm */ \
static __inline struct type * \
name##_SPLAY_FIND(struct name *head, struct type *elm) \
{ \
    if (SPLAY_EMPTY(head)) \
        return(NULL); \
    name##_SPLAY(head, elm); \
    if ((cmp)(elm, (head)->sph_root) == 0) \
        return (head->sph_root); \
    return (NULL); \
} \
\
static __inline _sys_tree_h_unused struct type * \
name##_SPLAY_NEXT(struct name *head, struct type *elm) \
{ \
    name##_SPLAY(head, elm); \
    if (SPLAY_RIGHT(elm, field) != NULL) { \
        elm = SPLAY_RIGHT(elm, field); \
        while (SPLAY_LEFT(elm, field) != NULL) { \
            elm = SPLAY_LEFT(elm, field); \
        } \
    } \
    else \
        elm = NULL; \
    return (elm); \
} \
\

```

```

static _sys_tree_h_unused __inline struct type * \
name##_SPLAY_MIN_MAX(struct name *head, int val) \
{ \
    name##_SPLAY_MINMAX(head, val); \
    return (SPLAY_ROOT(head)); \
}

/* Main splay operation.
 * Moves node close to the key of elm to top
 */
#define SPLAY_GENERATE(name, type, field, cmp) \
struct type * \
name##_SPLAY_INSERT(struct name *head, struct type *elm) \
{ \
    if (SPLAY_EMPTY(head)) { \
        SPLAY_LEFT(elm, field) = SPLAY_RIGHT(elm, field) = NULL; \
    } else { \
        int __comp; \
        name##_SPLAY(head, elm); \
        __comp = (cmp)(elm, (head)->sph_root); \
        if (__comp < 0) { \
            SPLAY_LEFT(elm, field) = SPLAY_LEFT((head)->sph_root, field); \
            SPLAY_RIGHT(elm, field) = (head)->sph_root; \
            SPLAY_LEFT((head)->sph_root, field) = NULL; \
        } else if (__comp > 0) { \
            SPLAY_RIGHT(elm, \
field) = SPLAY_RIGHT((head)->sph_root, field); \
            SPLAY_LEFT(elm, field) = (head)->sph_root; \
            SPLAY_RIGHT((head)->sph_root, field) = NULL; \
        } else \
            return ((head)->sph_root); \
    } \
    (head)->sph_root = (elm); \
    return (NULL); \
} \
\
struct type * \
name##_SPLAY_REMOVE(struct name *head, struct type *elm) \
{ \
    struct type *__tmp; \
    if (SPLAY_EMPTY(head)) \
        return (NULL); \
    name##_SPLAY(head, elm); \
    if ((cmp)(elm, (head)->sph_root) == 0) { \
        if (SPLAY_LEFT((head)->sph_root, field) == NULL) { \
            (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field); \
        } else { \
            __tmp = SPLAY_RIGHT((head)->sph_root, field); \

```

```

(head)->sph_root = SPLAY_LEFT((head)->sph_root, field);\
name##_SPLAY(head, elm); \
SPLAY_RIGHT((head)->sph_root, field) = __tmp; \
} \
return (elm); \
} \
return (NULL); \
} \
\
void \
name##_SPLAY(struct
name *head, struct type *elm) \
{ \
struct type __node, *__left, *__right, *__tmp; \
int __comp; \
\
SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
__left = __right = &__node; \
\
while ((__comp = (cmp)(elm, (head)->sph_root)) != 0) { \
if (__comp < 0) { \
__tmp = SPLAY_LEFT((head)->sph_root, field);\
if (__tmp == NULL) \
break; \
if ((cmp)(elm, __tmp) < 0){ \
SPLAY_ROTATE_RIGHT(head, __tmp, field);\
if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
break; \
} \
SPLAY_LINKLEFT(head, __right, field); \
} else if (__comp > 0) { \
__tmp = SPLAY_RIGHT((head)->sph_root, field);\
if (__tmp == NULL) \
break; \
if ((cmp)(elm, __tmp) > 0){ \
SPLAY_ROTATE_LEFT(head, __tmp, field);\
if (SPLAY_RIGHT((head)->sph_root, field) == NULL)\
break; \
} \
SPLAY_LINKRIGHT(head, __left, field); \
} \
} \
SPLAY_ASSEMBLE(head,
&__node, __left, __right, field); \
} \
\
/* Splay with either the minimum or the maximum element \
* Used to find minimum or maximum element in tree. \

```



```

*/
\
void name##_SPLAY_MINMAX(struct name *head, int __comp) \
{
\
struct type __node, *__left, *__right, *__tmp; \
\
SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
__left = __right = &__node; \
\
while (1) { \
if (__comp < 0) { \
__tmp = SPLAY_LEFT((head)->sph_root, field);\
if (__tmp == NULL) \
break; \
if (__comp < 0){ \
SPLAY_ROTATE_RIGHT(head, __tmp, field);\
if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
break; \
} \
SPLAY_LINKLEFT(head, __right, field); \
} else if (__comp > 0) { \
__tmp = SPLAY_RIGHT((head)->sph_root, field);\
if (__tmp == NULL) \
break; \
if (__comp > 0) { \
SPLAY_ROTATE_LEFT(head, __tmp, field);\
if (SPLAY_RIGHT((head)->sph_root, \
field) == NULL)\
break; \
} \
SPLAY_LINKRIGHT(head, __left, field); \
} \
} \
SPLAY_ASSEMBLE(head, &__node, __left, __right, field); \
}

```

```

#define SPLAY_NEGINF -1

```

```

#define SPLAY_INF 1

```

```

#define SPLAY_INSERT(name, x, y) name##_SPLAY_INSERT(x, y)

```

```

#define SPLAY_REMOVE(name, x, y) name##_SPLAY_REMOVE(x, y)

```

```

#define SPLAY_FIND(name, x, y) name##_SPLAY_FIND(x, y)

```

```

#define SPLAY_NEXT(name, x, y) name##_SPLAY_NEXT(x, y)

```

```

#define SPLAY_MIN(name, x) (SPLAY_EMPTY(x) ? NULL \

```

```

: name##_SPLAY_MIN_MAX(x, SPLAY_NEGINF))

```

```

#define SPLAY_MAX(name, x) (SPLAY_EMPTY(x) ? NULL \

```

```

: name##_SPLAY_MIN_MAX(x, SPLAY_INF))

```

```

#define SPLAY_FOREACH(x, name, head) \

```

```

for ((x) = SPLAY_MIN(name, head); \
    (x) != NULL; \
    (x) = SPLAY_NEXT(name, head, x))

/* Macros that define a red-black tree */
#define RB_HEAD(name, type) \
struct name { \
    struct type *rbh_root; /* root of the tree */ \
}

#define RB_INITIALIZER(root) \
{ \
    NULL \
}

#define RB_INIT(root) do { \
    (root)->rbh_root = NULL; \
} while (/*CONSTCOND*/ 0)

#define RB_BLACK 0
#define RB_RED 1
#define RB_ENTRY(type) \
struct { \
    struct type *rbe_left; /* left element */ \
    struct type *rbe_right; /* right element */ \
    struct type *rbe_parent; /* parent element */ \
    int rbe_color; /* node color */ \
}

#define RB_LEFT(elm, field) (elm)->field.rbe_left
#define RB_RIGHT(elm, field) (elm)->field.rbe_right
#define RB_PARENT(elm, field) (elm)->field.rbe_parent
#define RB_COLOR(elm, field) (elm)->field.rbe_color
#define RB_ROOT(head) (head)->rbh_root
#define RB_EMPTY(head) (RB_ROOT(head) == NULL)

#define RB_SET(elm, parent, field) do { \
    RB_PARENT(elm, field) = parent; \
    RB_LEFT(elm, field) = RB_RIGHT(elm, field) = NULL; \
    RB_COLOR(elm, field) = RB_RED; \
} while (/*CONSTCOND*/ 0)

#define RB_SET_BLACKRED(black, red, field) do { \
    RB_COLOR(black, field) = RB_BLACK; \
    RB_COLOR(red, field) \
    = RB_RED; \
} while (/*CONSTCOND*/ 0)

#ifndef RB_AUGMENT

```

```

#define RB_AUGMENT(x) do { } while (/*CONSTCOND*/ 0)
#endif

#define RB_ROTATE_LEFT(head, elm, tmp, field) do { \
    (tmp) = RB_RIGHT(elm, field); \
    if ((RB_RIGHT(elm, field) = RB_LEFT(tmp, field)) != NULL) { \
        RB_PARENT(RB_LEFT(tmp, field), field) = (elm); \
    } \
    RB_AUGMENT(elm); \
    if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
        if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
            RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
        else \
            RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
    } else \
        (head)->rbh_root = (tmp); \
    RB_LEFT(tmp, field) = (elm); \
    RB_PARENT(elm, field) = (tmp); \
    RB_AUGMENT(tmp); \
    if ((RB_PARENT(tmp, field))) \
        RB_AUGMENT(RB_PARENT(tmp, field)); \
} while (/*CONSTCOND*/ 0)

#define RB_ROTATE_RIGHT(head, elm, tmp, field) do { \
    (tmp) = RB_LEFT(elm, field); \
    if ((RB_LEFT(elm, field) = RB_RIGHT(tmp, field)) \
        != NULL) { \
        RB_PARENT(RB_RIGHT(tmp, field), field) = (elm); \
    } \
    RB_AUGMENT(elm); \
    if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
        if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
            RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
        else \
            RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
    } else \
        (head)->rbh_root = (tmp); \
    RB_RIGHT(tmp, field) = (elm); \
    RB_PARENT(elm, field) = (tmp); \
    RB_AUGMENT(tmp); \
    if ((RB_PARENT(tmp, field))) \
        RB_AUGMENT(RB_PARENT(tmp, field)); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */
#define RB_PROTOTYPE(name, type, field, cmp) \
    RB_PROTOTYPE_INTERNAL(name, type, field, cmp,)
#define RB_PROTOTYPE_STATIC(name, type, field, cmp) \

```

```

RB_PROTOTYPE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_PROTOTYPE_INTERNAL(name, type, field, cmp, attr) \
attr void name##_RB_INSERT_COLOR(struct name *, struct
type *); \
attr void name##_RB_REMOVE_COLOR(struct name *, struct type *, struct type *);\
attr struct type *name##_RB_REMOVE(struct name *, struct type *); \
attr struct type *name##_RB_INSERT(struct name *, struct type *); \
attr struct type *name##_RB_FIND(struct name *, struct type *); \
attr struct type *name##_RB_NFIND(struct name *, struct type *); \
attr struct type *name##_RB_NEXT(struct type *); \
attr struct type *name##_RB_PREV(struct type *); \
attr struct type *name##_RB_MINMAX(struct name *, int); \
\

/* Main rb operation.
* Moves node close to the key of elm to top
*/
#define RB_GENERATE(name, type, field, cmp) \
RB_GENERATE_INTERNAL(name, type, field, cmp,)
#define RB_GENERATE_STATIC(name, type, field, cmp) \
RB_GENERATE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_GENERATE_INTERNAL(name, type, field, cmp, attr) \
attr void \
name##_RB_INSERT_COLOR(struct name *head, struct type *elm) \
{ \
struct type
*parent, *gparent, *tmp; \
while ((parent = RB_PARENT(elm, field)) != NULL && \
RB_COLOR(parent, field) == RB_RED) { \
gparent = RB_PARENT(parent, field); \
if (parent == RB_LEFT(gparent, field)) { \
tmp = RB_RIGHT(gparent, field); \
if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
RB_COLOR(tmp, field) = RB_BLACK; \
RB_SET_BLACKRED(parent, gparent, field);\
elm = gparent; \
continue; \
} \
if (RB_RIGHT(parent, field) == elm) { \
RB_ROTATE_LEFT(head, parent, tmp, field);\
tmp = parent; \
parent = elm; \
elm = tmp; \
} \
RB_SET_BLACKRED(parent, gparent, field); \
RB_ROTATE_RIGHT(head, gparent, tmp, field); \
} else { \
tmp = RB_LEFT(gparent, field); \

```

```

if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
    RB_COLOR(tmp, field) = RB_BLACK; \
    RB_SET_BLACKRED(parent, gparent, field);\
    elm = gparent; \
    continue; \
} \
if (RB_LEFT(parent, field) == elm)
{ \
    RB_ROTATE_RIGHT(head, parent, tmp, field);\
    tmp = parent; \
    parent = elm; \
    elm = tmp; \
} \
RB_SET_BLACKRED(parent, gparent, field); \
RB_ROTATE_LEFT(head, gparent, tmp, field);\
} \
} \
RB_COLOR(head->rbh_root, field) = RB_BLACK; \
} \
\
attr void \
name##_RB_REMOVE_COLOR(struct name *head, struct type *parent, struct type *elm) \
{ \
    struct type *tmp; \
    while ((elm == NULL || RB_COLOR(elm, field) == RB_BLACK) && \
        elm != RB_ROOT(head)) { \
        if (RB_LEFT(parent, field) == elm) { \
            tmp = RB_RIGHT(parent, field); \
            if (RB_COLOR(tmp, field) == RB_RED) { \
                RB_SET_BLACKRED(tmp, parent, field); \
                RB_ROTATE_LEFT(head, parent, tmp, field);\
                tmp = RB_RIGHT(parent, field); \
            } \
            if ((RB_LEFT(tmp, field) == NULL || \
                RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) && \
                (RB_RIGHT(tmp, field) == NULL || \
                RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK)) {\
                RB_COLOR(tmp, field) = RB_RED; \
                elm = parent; \
                parent = RB_PARENT(elm, field); \
            } else { \
                if (RB_RIGHT(tmp, field) == NULL || \
                    RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK) {\
                    struct type *oleft; \
                    if ((oleft = RB_LEFT(tmp, field)) \
                        != NULL) \
                        RB_COLOR(oleft, field) = RB_BLACK;\

```

```

RB_COLOR(tmp, field) = RB_RED; \
RB_ROTATE_RIGHT(head, tmp, oleft, field);\
tmp = RB_RIGHT(parent, field); \
} \
RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
RB_COLOR(parent, field) = RB_BLACK; \
if (RB_RIGHT(tmp, field)) \
    RB_COLOR(RB_RIGHT(tmp, field), field) = RB_BLACK;\
RB_ROTATE_LEFT(head, parent, tmp, field);\
elm = RB_ROOT(head); \
break; \
} \
} else { \
    tmp = RB_LEFT(parent, field); \
    if (RB_COLOR(tmp, field) == RB_RED) { \
        RB_SET_BLACKRED(tmp, parent, field); \
        RB_ROTATE_RIGHT(head, parent, tmp, field);\
        tmp
= RB_LEFT(parent, field); \
    } \
    if ((RB_LEFT(tmp, field) == NULL || \
        RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) &&\
        (RB_RIGHT(tmp, field) == NULL || \
        RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK)) {\
        RB_COLOR(tmp, field) = RB_RED; \
        elm = parent; \
        parent = RB_PARENT(elm, field); \
    } else { \
        if (RB_LEFT(tmp, field) == NULL || \
            RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) {\
            struct type *oright; \
            if ((oright = RB_RIGHT(tmp, field)) \
                != NULL) \
                RB_COLOR(oright, field) = RB_BLACK;\
            RB_COLOR(tmp, field) = RB_RED; \
            RB_ROTATE_LEFT(head, tmp, oright, field);\
            tmp = RB_LEFT(parent, field); \
        } \
        RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
        RB_COLOR(parent, field) = RB_BLACK; \
        if (RB_LEFT(tmp, field)) \
            RB_COLOR(RB_LEFT(tmp, field), field) = RB_BLACK;\
        RB_ROTATE_RIGHT(head, parent, tmp, field);\
        elm = RB_ROOT(head); \
        break; \
    } \
} \
} \
} \

```

```

if
(elm)    \
    RB_COLOR(elm, field) = RB_BLACK; \
}      \
    \
attr struct type *    \
name##_RB_REMOVE(struct name *head, struct type *elm) \
{      \
    struct type *child, *parent, *old = elm; \
    int color;    \
    if (RB_LEFT(elm, field) == NULL) \
        child = RB_RIGHT(elm, field); \
    else if (RB_RIGHT(elm, field) == NULL) \
        child = RB_LEFT(elm, field); \
    else {      \
        struct type *left; \
        elm = RB_RIGHT(elm, field); \
        while ((left = RB_LEFT(elm, field)) != NULL) \
            elm = left; \
        child = RB_RIGHT(elm, field); \
        parent = RB_PARENT(elm, field); \
        color = RB_COLOR(elm, field); \
        if (child) \
            RB_PARENT(child, field) = parent; \
        if (parent) { \
            if (RB_LEFT(parent, field) == elm) \
                RB_LEFT(parent, field) = child; \
            else \
                RB_RIGHT(parent, field) = child; \
            RB_AUGMENT(parent); \
        } else \
            RB_ROOT(head)
    = child; \
    if (RB_PARENT(elm, field) == old) \
        parent = elm; \
    (elm)->field = (old)->field; \
    if (RB_PARENT(old, field)) { \
        if (RB_LEFT(RB_PARENT(old, field), field) == old)\
            RB_LEFT(RB_PARENT(old, field), field) = elm;\
        else \
            RB_RIGHT(RB_PARENT(old, field), field) = elm;\
        RB_AUGMENT(RB_PARENT(old, field)); \
    } else \
        RB_ROOT(head) = elm; \
    RB_PARENT(RB_LEFT(old, field), field) = elm; \
    if (RB_RIGHT(old, field)) \
        RB_PARENT(RB_RIGHT(old, field), field) = elm; \
    if (parent) { \

```

```

    left = parent; \
    do { \
        RB_AUGMENT(left); \
    } while ((left = RB_PARENT(left, field)) != NULL); \
} \
goto color; \
} \
parent = RB_PARENT(elm, field); \
color = RB_COLOR(elm, field); \
if (child) \
    RB_PARENT(child, field) = parent; \
if (parent) { \
    if (RB_LEFT(parent, field) == elm) \
        RB_LEFT(parent, field) = child; \
    else \
        RB_RIGHT(parent,
field) = child; \
    RB_AUGMENT(parent); \
} else \
    RB_ROOT(head) = child; \
color: \
if (color == RB_BLACK) \
    name##_RB_REMOVE_COLOR(head, parent, child); \
return (old); \
} \
\
/* Inserts a node into the RB tree */ \
attr struct type * \
name##_RB_INSERT(struct name *head, struct type *elm) \
{ \
    struct type *tmp; \
    struct type *parent = NULL; \
    int comp = 0; \
    tmp = RB_ROOT(head); \
    while (tmp) { \
        parent = tmp; \
        comp = (cmp)(elm, parent); \
        if (comp < 0) \
            tmp = RB_LEFT(tmp, field); \
        else if (comp > 0) \
            tmp = RB_RIGHT(tmp, field); \
        else \
            return (tmp); \
    } \
    RB_SET(elm, parent, field); \
    if (parent != NULL) { \
        if (comp < 0) \
            RB_LEFT(parent, field) = elm; \

```



```

else    \
    RB_RIGHT(parent, field) = elm; \
    RB_AUGMENT(parent); \
} else    \
    RB_ROOT(head) = elm; \
name##_RB_INSERT_COLOR(head,
elm); \
return (NULL); \
} \
\

/* Finds the node with the same key as elm */ \
attr struct type * \
name##_RB_FIND(struct name *head, struct type *elm) \
{ \
    struct type *tmp = RB_ROOT(head); \
    int comp; \
    while (tmp) { \
        comp = cmp(elm, tmp); \
        if (comp < 0) \
            tmp = RB_LEFT(tmp, field); \
        else if (comp > 0) \
            tmp = RB_RIGHT(tmp, field); \
        else \
            return (tmp); \
    } \
    return (NULL); \
} \
\

/* Finds the first node greater than or equal to the search key */ \
attr struct type * \
name##_RB_NFIND(struct name *head, struct type *elm) \
{ \
    struct type *tmp = RB_ROOT(head); \
    struct type *res = NULL; \
    int comp; \
    while (tmp) { \
        comp = cmp(elm, tmp); \
        if (comp < 0) { \
            res = tmp; \
            tmp = RB_LEFT(tmp, field); \
        } \
        else if (comp > 0) \
            tmp =
RB_RIGHT(tmp, field); \
        else \
            return (tmp); \
    } \
    return (res); \
}

```

```

}    \
    \
/* ARGSUSED */    \
attr struct type *    \
name##_RB_NEXT(struct type *elm)    \
{    \
    if (RB_RIGHT(elm, field)) {    \
        elm = RB_RIGHT(elm, field);    \
        while (RB_LEFT(elm, field))    \
            elm = RB_LEFT(elm, field);    \
    } else {    \
        if (RB_PARENT(elm, field) &&    \
            (elm == RB_LEFT(RB_PARENT(elm, field), field)))    \
            elm = RB_PARENT(elm, field);    \
        else {    \
            while (RB_PARENT(elm, field) &&    \
                (elm == RB_RIGHT(RB_PARENT(elm, field), field)))    \
                elm = RB_PARENT(elm, field);    \
        }    \
    }    \
    return (elm);    \
}    \
    \
/* ARGSUSED */    \
attr struct type *    \
name##_RB_PREV(struct type *elm)    \
{    \
    if (RB_LEFT(elm, field)) {    \
        elm = RB_LEFT(elm, field);    \
        while (RB_RIGHT(elm, field))    \
            elm = RB_RIGHT(elm, field);    \
    }    \
    else {    \
        if (RB_PARENT(elm, field) &&    \
            (elm == RB_RIGHT(RB_PARENT(elm, field), field)))    \
            elm = RB_PARENT(elm, field);    \
        else {    \
            while (RB_PARENT(elm, field) &&    \
                (elm == RB_LEFT(RB_PARENT(elm, field), field)))    \
                elm = RB_PARENT(elm, field);    \
        }    \
    }    \
    return (elm);    \
}    \
    \
attr struct type *    \

```

```

name##_RB_MINMAX(struct name *head, int val) \
{ \
    struct type *tmp = RB_ROOT(head); \
    struct type *parent = NULL; \
    while (tmp) { \
        parent = tmp; \
        if (val < 0) \
            tmp = RB_LEFT(tmp, field); \
        else \
            tmp = RB_RIGHT(tmp, field); \
    } \
    return (parent); \
}

#define RB_NEGINF -1
#define RB_INF 1

#define RB_INSERT(name, x, y) name##_RB_INSERT(x, y)
#define RB_REMOVE(name, x, y) name##_RB_REMOVE(x, y)
#define RB_FIND(name, x, y) name##_RB_FIND(x, y)
#define RB_NFIND(name, x, y) name##_RB_NFIND(x, \
y)
#define RB_NEXT(name, x, y) name##_RB_NEXT(y)
#define RB_PREV(name, x, y) name##_RB_PREV(y)
#define RB_MIN(name, x) name##_RB_MINMAX(x, RB_NEGINF)
#define RB_MAX(name, x) name##_RB_MINMAX(x, RB_INF)

#define RB_FOREACH(x, name, head) \
for ((x) = RB_MIN(name, head); \
(x) != NULL; \
(x) = name##_RB_NEXT(x))

#define RB_FOREACH_FROM(x, name, y) \
for ((x) = (y); \
((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
(x) = (y))

#define RB_FOREACH_SAFE(x, name, head, y) \
for ((x) = RB_MIN(name, head); \
((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
(x) = (y))

#define RB_FOREACH_REVERSE(x, name, head) \
for ((x) = RB_MAX(name, head); \
(x) != NULL; \
(x) = name##_RB_PREV(x))

#define RB_FOREACH_REVERSE_FROM(x, name, y) \

```

```

for ((x) = (y);    \
    ((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
    (x) = (y))

#define RB_FOREACH_REVERSE_SAFE(x, name,
head, y) \
for ((x) = RB_MAX(name, head);    \
    ((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
    (x) = (y))

#endif /* _SYS_TREE_H_ */
pkgname=linux-headers
pkgver=6.14.2 # Follow the latest Linux stable
_kernver=${pkgver%.*}
pkgrel=0
pkgdesc="Linux system headers"
url="https://kernel.org/"
arch="all" # Headers are architecture specific
license="GPL-2.0-only"
makedepends="perl"
options="!check !dbg !strip !tracedeps !archcheck"
source="https://kernel.org/pub/linux/kernel/v6.x/linux-${_kernver}.tar.xz
revert-broken-uapi.patch
0003-remove-inclusion-of-sysinfo.h-in-kernel.h.patch

```

## 1.131 autocommand 2.2.2

### 1.131.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<https://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

#### 0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License,

other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

#### 1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

#### 2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

- a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or
- b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

#### 3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from

a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

- a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the object code with a copy of the GNU GPL and this license document.

#### 4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

- a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the Combined Work with a copy of the GNU GPL and this license document.
- c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.
- d) Do one of the following:

- 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

- 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer

system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

## 5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
- b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

## 6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you

received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

# Copyright 2014-2015 Nathan West

#

# This file is part of autocommand.

#

# autocommand is free software: you can redistribute it and/or modify

# it under the terms of the GNU Lesser General Public License as published by

# the Free Software Foundation, either version 3 of the License, or

# (at your option) any later version.

#

# autocommand is distributed in the hope that it will be useful,

# but WITHOUT ANY WARRANTY; without even the implied warranty of

# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

# GNU Lesser General Public License for more details.

#

# You should have received a copy of the GNU Lesser General Public License

# along with autocommand. If not, see <<http://www.gnu.org/licenses/>>.

## 1.132 jaraco.text 3.12.1

### 1.132.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN



THE SOFTWARE.  
MIT

## 1.133 python-certifi 2024.08.30

### 1.133.1 Available under license :

This package contains a modified version of ca-bundle.crt:

ca-bundle.crt -- Bundle of CA Root Certificates

This is a bundle of X.509 certificates of public Certificate Authorities (CA). These were automatically extracted from Mozilla's root certificates file (certdata.txt). This file can be found in the mozilla source tree:  
<https://hg.mozilla.org/mozilla-central/file/tip/security/nss/lib/ckfw/builtins/certdata.txt>  
It contains the certificates in PEM format and therefore can be directly used with curl / libcurl / php\_curl, or with an Apache+mod\_ssl webserver for SSL client authentication.  
Just configure this file as the SSLCACertificateFile.#

\*\*\*\*\* BEGIN LICENSE BLOCK \*\*\*\*\*

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

\*\*\*\*\* END LICENSE BLOCK \*\*\*\*\*

@(#) \$RCSfile: certdata.txt,v \$ \$Revision: 1.80 \$ \$Date: 2011/11/03 15:11:58 \$

## 1.134 dhcping 1.2-r3

### 1.134.1 Available under license :

Non Maintainer Upload of this Package

-----  
If you plan to work on an NMU for this package, read the following closely. It can save you and me some grief.

1. At first, contact the maintainer (i.e. send a mail to [joey@debian.org](mailto:joey@debian.org), do not cc or bounce a mail, send a plain mail, not copied to any mailing list or the BTS) and ask about the status of the bug you are considering to work on.
2. In this mail include all information relevant for this problem, i.e. include a description of the bug and not only its bug number.
3. If the maintainer is not able or willing to fix the problem or

does not respond within four days, continue with step 4.

4. Work on the bug and prepare a patch. Do not upload into the Debian archive.
5. Send the entire patch, together with enough explanations, to the maintainer for reviewing and ask him for permission of an NMU using this patch.
6. IF AND ONLY IF the maintainer approves the patch (or doesn't respond within four days), upload the NMU to the incoming directory and send the patch to the BTS. If the NMU is not approved, go back to 4. or add the NMU to your homepage, but do not upload it to the Debian archive.
7. Properly sized and well-written patches sent to the BTS are always appreciated, even if they are rejected later. They demonstrate a potential solution which could probably improved into a real solution.
8. NEVER change the way a package is maintained in an NMU, i.e. don't remove `dh_*` stuff or switch to `dh_*` respectively. This rule applies to all NMU's, not only to an NMU for this package.

These rules always apply. They even apply if somebody declares NMUs as ok and reduces regular NMU rules to a delay of zero days. Unless I'm on vacation or on a show I am reachable via mail, so there is hardly a reason not to contact me.

This is the Debian GNU prepackaged version of `dhcping`, an administrative tool for finding out if a DHCP server is still up and running.

The source of the Debian package is managed through CVS. It is publically available at `<http://cvs.infodrom.org/dhcping/?cvsroot=debian>` or as `<:pserver:anonymous@cvs.infodrom.org/var/cvs/debian/dhcping/>`.

This package was put together by Martin Schulze `<joey@debian.org>`, from sources obtained from:

<http://www.mavetju.org/download/dhcping-1.2.tar.gz>

`dhcping` has been released under the BSD license as quoted below.

Copyright (c) 2000, 2001, 2002 Edwin Groothuis, `edwin@mavetju.org`  
All rights reserved.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.135 readline 8.2.13-r1

### 1.135.1 Available under license :

@c The GNU Free Documentation License.  
@center Version 1.3, 3 November 2008

@c This file is intended to be included within another document,  
@c hence no sectioning command or @node.

@display  
Copyright @copyright{} 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.  
@uref{http://fsf.org/}

Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.  
@end display

@enumerate 0  
@item  
PREAMBLE

The purpose of this License is to make a manual, textbook, or other  
functional and useful document @dfn{free} in the sense of freedom: to  
assure everyone the effective freedom to copy and redistribute it,  
with or without modifying it, either commercially or noncommercially.  
Secondarily, this License preserves for the author and publisher a way

to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of ``copyleft'', which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

@item

## APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The ``Document'', below, refers

to any such manual or work. Any member of the public is a licensee, and is addressed as ``you''. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A ``Modified Version'' of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A ``Secondary Section'' is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The ``Invariant Sections'' are certain Secondary Sections whose titles

are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The ``Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A ``Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not ``Transparent" is called ``Opaque".

Examples of suitable formats for Transparent copies include plain `@sc{ascii}` without markup, Texinfo input format, `La@TeX{}` input format, `@acronym{SGML}` or `@acronym{XML}` using a publicly available `@acronym{DTD}`, and standard-conforming simple `@acronym{HTML}`, PostScript or `@acronym{PDF}` designed for human modification. Examples of transparent image formats include `@acronym{PNG}`, `@acronym{XCF}` and `@acronym{JPG}`. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, `@acronym{SGML}` or `@acronym{XML}` for which the `@acronym{DTD}` and/or processing tools are not generally available, and the machine-generated `@acronym{HTML}`, PostScript or `@acronym{PDF}` produced by some word processors for output purposes only.

The ``Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, ``Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The ``publisher" means any person or entity that distributes copies

of the Document to the public.

A section ``Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in

another language. (Here XYZ stands for a specific section name mentioned below, such as ``Acknowledgements", ``Dedications", ``Endorsements", or ``History".) To ``Preserve the Title" of such a section when you modify the Document means that it remains a section ``Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

@item

#### VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever

to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

@item

#### COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible.

You may add other material on the covers in addition.

Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that

this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

@item

## MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

@enumerate A

@item

Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions

(which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

@item

List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

@item

State on the Title page the name of the publisher of the Modified Version, as the publisher.

@item

Preserve all the copyright notices of the Document.

@item

Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

@item

Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

@item

Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

@item

Include an unaltered copy of this License.

@item

Preserve the section Entitled ``History'', Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled ``History'' in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

@item

Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the ``History'' section. You may omit a network location for a work that was published at least

four years before the Document itself, or if the original publisher of the version it refers to gives permission.



@item

For any section Entitled ``Acknowledgements" or ``Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

@item

Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

@item

Delete any section Entitled ``Endorsements". Such a section may not be included in the Modified Version.

@item

Do not retitle any existing section to be Entitled ``Endorsements" or to conflict in title with any Invariant Section.

@item

Preserve any Warranty Disclaimers.

@end enumerate

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document,  
you may at your option designate some or all  
of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled ``Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties---for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of,  
you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

@item

## COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled ``History" in the various original documents, forming one section Entitled ``History"; likewise combine any sections Entitled ``Acknowledgements", and any sections Entitled ``Dedications". You must delete all sections Entitled ``Endorsements."

@item

## COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

@item

## AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an ``aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

@item

#### TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled ``Acknowledgements", ``Dedications", or ``History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

@item

#### TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

@item

#### FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See [{http://www.gnu.org/copyleft/}](http://www.gnu.org/copyleft/).

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License ``or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

@item

#### RELICENSING

``Massive Multiauthor Collaboration Site" (or ``MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A ``Massive Multiauthor Collaboration" (or ``MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

``CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

``Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is ``eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

@end enumerate

@page

@heading ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

@smallexample

@group

Copyright (C) @var{year} @var{your name}.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled ``GNU Free Documentation License".

@end group

@end smallexample

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the ``with@dots{ }Texts." line with this:

@smallexample

@group

with the Invariant Sections being @var{list their titles}, with  
the Front-Cover

Texts being @var{list}, and with the Back-Cover Texts  
being @var{list}.

@end group

@end smallexample

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

@c Local Variables:

@c ispell-local-pdict: "ispell-dict"

@c End:

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

## Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed

to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it,

either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies  
of the Program's

source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)



These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to

control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent

license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.

##### EXCEPT WHEN

OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN

IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and  
`show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

# GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

## Preamble

The GNU General Public License is a free, copyleft license for  
software and other kinds of works.

The licenses for most software and other practical works are designed  
to take away your freedom to share and change the works. By contrast,  
the GNU General Public License is intended to guarantee your freedom to  
share and change all versions of a program--to make sure it remains free  
software for all its users. We, the Free Software Foundation, use the  
GNU General Public License for most of our software; it applies also to  
any other work released this way by its authors. You can apply it to  
your programs, too.

When we speak of free software, we are referring to  
freedom, not  
price. Our General Public Licenses are designed to make sure that you  
have the freedom to distribute copies of free software (and charge for  
them if you wish), that you receive source code or can get it if you  
want it, that you can change the software or use pieces of it in new  
free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you  
these rights or asking you to surrender the rights. Therefore, you have  
certain responsibilities if you distribute copies of the software, or if  
you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether  
gratis or for a fee, you must pass on to the recipients the same  
freedoms that you received. You must make sure that they, too, receive  
or can get the source code. And you must show them these terms so they  
know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1)

assert copyright on the software, and (2) offer you this License  
giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains  
that there is no warranty for this free software. For both users' and  
authors' sake, the GPL requires that modified versions be marked as

changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS

### 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without

permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

#### 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The

"System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally



available free

programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

## 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered

by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed;

section 10

makes it unnecessary.

## 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or

similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

#### 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

#### 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

## 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated

place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the

User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

## 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or

limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on

those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

## 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or

modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

#### 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

#### 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that

transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights

granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

#### 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this



License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

## 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may

not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

### 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

### 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

### 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS"

WITHOUT WARRANTY

OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17.

Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by

the Free Software Foundation, either version 3 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.

You should have received a copy of the GNU General Public License  
along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short  
notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>  
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate  
parts of the General Public License. Of course, your program's commands  
might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school,  
if any, to sign a "copyright disclaimer" for the program, if necessary.  
For more information on this, and how to apply and follow the GNU GPL, see  
<http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program  
into proprietary programs. If your program is a subroutine library, you  
may consider it more useful to permit linking proprietary applications with  
the library. If this is what you want to  
do, use the GNU Lesser General  
Public License instead of this License. But first, please read  
<http://www.gnu.org/philosophy/why-not-lgpl.html>.

# 1.136 toml 2.0.1

## 1.136.1 Available under license :

The MIT License (MIT)

Copyright (c) 2018 TOML authors

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR

IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2021 Taneli Hukkinen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

SPDX-License-Identifier: MIT

SPDX-FileCopyrightText: 2021 Taneli Hukkinen

Licensed to PSF under a Contributor Agreement.

The MIT License

Copyright (c) 2021 Pradyun Gedam and Contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Portions of the Software include code copyright (c) Rebecca Turner.  
MIT License

Copyright (c) 2020 Bob Fang (bob.fang.london@gmail.com)  
Copyright (c) 2021 Taneli Hukkinen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.137 liboping 1.10.0-r2

### 1.137.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE  
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts  
as the successor of the GNU Library Public License, version 2, hence  
the version number 2.1.]

## Preamble

The licenses for most software are designed to take away your  
freedom to share and change it. By contrast, the GNU General Public  
Licenses are intended to guarantee your freedom to share and change  
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some  
specially designated software packages--typically libraries--of the  
Free Software Foundation and other authors who  
decide to use it. You  
can use it too, but we suggest you first think carefully about whether  
this license or the ordinary General Public License is the better  
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,  
not price. Our General Public Licenses are designed to make sure that  
you have the freedom to distribute copies of free software (and charge  
for this service if you wish); that you receive source code or can get  
it if you want it; that you can change the software and use pieces of  
it in new free programs; and that you are informed that you can do  
these things.

To protect your rights, we need to make restrictions that forbid  
distributors to deny you these rights or to ask you to surrender these  
rights. These restrictions translate to certain responsibilities for  
you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library,  
whether gratis  
or for a fee, you must give the recipients all the rights that we gave  
you. You must make sure that they, too, receive or can get the source  
code. If you link other code with the library, you must provide  
complete object files to the recipients, so that they can relink them  
with the library after making changes to the library and recompiling  
it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the  
library, and (2) we offer you this license, which gives you legal

permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting



the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

## GNU LESSER GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation

and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any

application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

### 3. You may opt to apply the terms of the ordinary GNU General Public

License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding

machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse

engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
- b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is

normally

distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the

Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.



## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!

# 1.138 libmd 1.1.0-r0

## 1.138.1 Available under license :

Format: <https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Files:

\*

Copyright:

Copyright 2009, 2011, 2016 Guillem Jover <guillem@hadrons.org>

License: BSD-3-clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:

include/sha2.h

src/sha2.c

Copyright:

Copyright 2000-2001, Aaron D. Gifford

All rights reserved.

License: BSD-3-clause-Aaron-D-Gifford

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce

the above copyright

notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

.  
THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTOR(S) ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:

include/rmd160.h

src/rmd160.c

Copyright:

Copyright 2001 Markus Friedl. All rights reserved.

License: BSD-2-clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

.  
THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:

src/md2.c

Copyright:

Copyright (c) 2001 The NetBSD Foundation, Inc.

All rights reserved.

.

This code is derived from software contributed to The NetBSD Foundation  
by Andrew Brown.

License: BSD-2-clause-NetBSD

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions  
are met:

1. Redistributions of source code must retain the above copyright  
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must  
reproduce the above copyright  
notice, this list of conditions and the following disclaimer in the  
documentation and/or other materials provided with the distribution.

.

THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS  
``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED  
TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR  
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS  
BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR  
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF  
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS  
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN  
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)  
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE  
POSSIBILITY OF SUCH DAMAGE.

Files:

man/rmd160.3

man/sha1.3

man/sha2.3

Copyright:

Copyright 1997, 2003, 2004 Todd C. Miller <Todd.Miller@courtesan.com>

License: ISC

Permission to use, copy, modify, and distribute this software for any  
purpose with or without fee is hereby granted, provided that the above  
copyright notice and this permission notice appear in all copies.

.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES  
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF  
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR  
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES  
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN  
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF  
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Files:

man/mdX.3

src/helper.c

Copyright:

Poul-Henning Kamp <phk@login.dkuug.dk>

License: Beerware

"THE BEER-WARE LICENSE" (Revision 42):

<phk@login.dkuug.dk> wrote this file. As long as you retain  
this notice you

can do whatever you want with this stuff. If we meet some day, and you think  
this stuff is worth it, you can buy me a beer in return. Poul-Henning Kamp

Files:

include/md4.h

src/md4.c

Copyright:

Colin Plumb

Todd C. Miller

License: public-domain-md4

This code implements the MD4 message-digest algorithm.

The algorithm is due to Ron Rivest. This code was  
written by Colin Plumb in 1993, no copyright is claimed.

This code is in the public domain; do with it what you wish.

Todd C. Miller modified the MD5 code to do MD4 based on RFC 1186.

Files:

include/md5.h

src/md5.c

Copyright:

Colin Plumb

License: public-domain-md5

This code implements the MD5 message-digest algorithm.

The algorithm is due to Ron Rivest. This code was  
written by Colin Plumb in 1993, no copyright is claimed.

This code is in the public domain; do with it what you wish.

Files:

include/sha1.h

src/sha1.c

Copyright:

Steve Reid <steve@edmweb.com>

License: public-domain-sha1

100% Public

Domain

# 1.139 iftop 0.17-r11

## 1.139.1 Available under license :

Authors:

Paul Warren <pdw@ex-parrot.com>

Chris Lightfoot <chris@ex-parrot.com>

Download: <http://www.ex-parrot.com/~pdw/iftop/download/>

Files: \*

Copyright:

(C) 2002-2004 Paul Warren <pdw@ex-parrot.com>

(C) 2002-2004 Chris Lightfoot <chris@ex-parrot.com>

License: GPL-2+

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

.

On Debian systems, the complete text of the GNU General Public License can be found in /usr/share/common-licenses/GPL-2 file.

Files: debian/\*

Copyright: 2005-2009 Daniel Baumann <daniel@debian.org>

License: GPL-2+

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

On Debian systems, the complete text of the GNU General Public License can be found in `/usr/share/common-licenses/GPL-2` file.

# 1.140 python-setuptools 80.9.0

## 1.140.1 Available under license :

The code contained in this directory was automatically generated using the following command:

```
python -m validate_pyproject.pre_compile --output-dir=setuptools/config/_validate_pyproject --enable-plugins
setuptools.distutils --very-verbose -t setuptools=setuptools/config/setuptools.schema.json -t
distutils=setuptools/config/distutils.schema.json
```

Please avoid changing it manually.

You can report issues or suggest changes directly to ``validate-pyproject`` (or to the relevant plugin repository)

- <https://github.com/abravalheri/validate-pyproject/issues>

\*\*\*

The following files include code from opensource projects (either as direct copies or modified versions):

- ``fastjsonschema_exceptions.py``:
  - project: ``fastjsonschema`` - licensed under BSD-3-Clause (<https://github.com/horejsek/python-fastjsonschema>)
- ``extra_validations.py`` and ``format.py``, ``error_reporting.py``:
  - project: ``validate-pyproject`` - licensed under MPL-2.0 (<https://github.com/abravalheri/validate-pyproject>)

Additionally

the following files are automatically generated by tools provided by the same projects:

- ``__init__.py``
- ``fastjsonschema_validations.py``

The relevant copyright notes and licenses are included below.

\*\*\*

`fastjsonschema`

=====

Copyright (c) 2018, Michal Horejsek

All rights reserved.

Redistribution and use in source and binary forms, with or without modification,  
are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this  
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this  
list of conditions and the following disclaimer in the documentation and/or  
other materials provided with the distribution.

Neither the name of the {organization} nor the names of its  
contributors may be used to endorse or promote products derived from  
this software without specific prior written permission.

THIS

SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND  
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED  
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE  
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR  
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES  
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;  
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON  
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT  
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS  
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

\*\*\*

`validate-pyproject`

=====

Mozilla Public License, version 2.0

## 1. Definitions

### 1.1. "Contributor"

means each individual or legal entity that creates, contributes to the



creation of, or owns Covered Software.

1.2.

"Contributor Version"

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"

means Covered Software of a particular Contributor.

1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"

means

a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. "Executable Form"

means any form of the work other than Source Code Form.

1.7. "Larger Work"

means

a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. "License"

means this document.

1.9. "Licensable"

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

#### 1.10. "Modifications"

means any of the following:

- a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or
- b. any new file in Source Code Form that contains any Covered Software.

#### 1.11. "Patent Claims" of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

#### 1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

#### 1.13. "Source Code Form"

means the form of the work preferred for making modifications.

#### 1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

### 2. License Grants and Conditions

#### 2.1.

##### Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- a. under intellectual property rights (other than patent or trademark)

Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

- b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

## 2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

## 2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or
- b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

## 2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this

License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

## 2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

## 2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

## 2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

## 3. Responsibilities

### 3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

### 3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

### 3.3. Distribution of a Larger Work

You may create

and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

### 3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

### 3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

## 4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

## 5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days

after You have come back into

compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

## 6. Disclaimer of Warranty

Covered Software is provided under this License on an "as is" basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

## 7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any

character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

## 8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

## 9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

## 10. Versions of the License

### 10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

### 10.2.

#### Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

### 10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

#### Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v.

2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <https://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

#### Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

MIT License

Copyright (c) 2012 Daniel Holth <dholth@fastmail.fm> and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.



THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING

FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<https://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

#### 0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the

Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

- a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the Combined Work with a copy of the GNU GPL and this license document.
- c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.
- d) Do one of the following:
  - 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.
  - 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.
- e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

## 5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library

facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
- b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

#### 6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

#### A. HISTORY OF THE SOFTWARE

=====

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see <https://www.cwi.nl>) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see <https://www.cnri.reston.va.us>) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations, which became Zope Corporation. In 2001, the Python Software Foundation (PSF, see <https://www.python.org/psf/>) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation was a sponsoring member of the PSF.

All

Python releases are Open Source (see <https://opensource.org> for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

| Release        | Derived from | Year      | Owner      | GPL-compatible? (1) |
|----------------|--------------|-----------|------------|---------------------|
| 0.9.0 thru 1.2 |              | 1991-1995 | CWI        | yes                 |
| 1.3 thru 1.5.2 | 1.2          | 1995-1999 | CNRI       | yes                 |
| 1.6            | 1.5.2        | 2000      | CNRI       | no                  |
| 2.0            | 1.6          | 2000      | BeOpen.com | no                  |
| 1.6.1          | 1.6          | 2001      | CNRI       | yes (2)             |
| 2.1            | 2.0+1.6.1    | 2001      | PSF        | no                  |
| 2.0.1          | 2.0+1.6.1    | 2001      | PSF        | yes                 |
| 2.1.1          | 2.1+2.0.1    | 2001      | PSF        | yes                 |
| 2.1.2          | 2.1.1        | 2002      | PSF        | yes                 |
| 2.1.3          | 2.1.2        | 2002      | PSF        | yes                 |
| 2.2 and above  | 2.1.1        | 2001-now  |            |                     |
| PSF            | yes          |           |            |                     |

Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

## B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

Python software and documentation are licensed under the Python Software Foundation License Version 2.

Starting with Python 3.8.6, examples, recipes, and other code in the documentation are dual licensed under the PSF License Version 2 and the Zero-Clause BSD license.

Some software incorporated into Python is under different licenses. The licenses are listed with code falling under that license.

### PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.
2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023 Python Software Foundation; All Rights Reserved" are retained in Python alone or in any derivative version prepared by Licensee.
3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.
4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.
5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON

FOR ANY

INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS  
A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON,  
OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material  
breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any  
relationship of agency, partnership, or joint venture between PSF and  
Licensee. This License Agreement does not grant permission to use PSF  
trademarks or trade name in a trademark sense to endorse or promote  
products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee  
agrees to be bound by the terms and conditions of this License  
Agreement.

#### BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

-----

#### BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an  
office at 160 Saratoga  
Avenue, Santa Clara, CA 95051, and the  
Individual or Organization ("Licensee") accessing and otherwise using  
this software in source or binary form and its associated  
documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License  
Agreement, BeOpen hereby grants Licensee a non-exclusive,  
royalty-free, world-wide license to reproduce, analyze, test, perform  
and/or display publicly, prepare derivative works, distribute, and  
otherwise use the Software alone or in any derivative version,  
provided, however, that the BeOpen Python License is retained in the  
Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS"  
basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR  
IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND  
DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS  
FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE  
WILL NOT  
INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE

SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at <http://www.pythonlabs.com/logos.html> may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

#### CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

-----

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This



Agreement may also be obtained from a proxy server on the internet using the following URL: <http://hdl.handle.net/1895.22/1013>".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

## CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

-----

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam,  
The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its  
documentation for any purpose and without fee is hereby granted,  
provided that the above copyright notice appear in all copies and that  
both that copyright notice and this permission notice appear in  
supporting documentation, and that the name of Stichting Mathematisch  
Centrum or CWI not be used in advertising or publicity pertaining to  
distribution of the software without specific, written prior  
permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO  
THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND  
FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE  
FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES  
WHATSOEVER

RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN  
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT  
OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

## ZERO-CLAUSE BSD LICENSE FOR CODE IN THE PYTHON DOCUMENTATION

-----

Permission to use, copy, modify, and/or distribute this software for any  
purpose with or without fee is hereby granted.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH  
REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY  
AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT,  
INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM  
LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR  
OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR  
PERFORMANCE OF THIS SOFTWARE.

UNRECOGNIZED LICENSE; MD5 sum: 525adedaf9a94edbb3e245cdd7b0f448

UNRECOGNIZED LICENSE; MD5 sum: 8f2937ac54e1c71b8f33618e9e805225

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

## 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner

or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices

stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

#### END OF TERMS AND CONDITIONS

This software is made available under the terms of *\*either\** of the licenses found in LICENSE.APACHE or LICENSE.BSD. Contributions to this software is made under the terms of *\*both\** these licenses.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN

THE SOFTWARE.

MIT License

Copyright (c) 2021 Taneli Hukkinen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2010-202x The platformdirs developers

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) Donald Stufft and individual contributors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2012 Erik Rose

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,

the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate

comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The following files include code from opensource projects (either as direct copies or modified versions):

- `setuptools.schema.json`, `distutils.schema.json`:
  - project: `validate-pyproject` - licensed under MPL-2.0 (<https://github.com/abravalheri/validate-pyproject>):

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <https://mozilla.org/MPL/2.0/>.

This is the MIT license: <http://www.opensource.org/licenses/mit-license.php>

Copyright (c) Alex Grnholm

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,

TORT OR  
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR  
OTHER  
DEALINGS IN THE SOFTWARE.

## 1.141 tomli 2.2.1

### 1.141.1 Available under license :

The MIT License (MIT)

Copyright (c) 2018 TOML authors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR

IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2021 Taneli Hukkinen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

SPDX-License-Identifier: MIT

SPDX-FileCopyrightText: 2021 Taneli Hukkinen

Licensed to PSF under a Contributor Agreement.

The MIT License

Copyright (c) 2021 Pradyun Gedam and Contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT

OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Portions of the Software include code copyright (c) Rebecca Turner.

MIT License

Copyright (c) 2020 Bob Fang (bob.fang.london@gmail.com)

Copyright (c) 2021 Taneli Hukkinen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND

NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR  
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN  
AN ACTION OF CONTRACT,  
TORT OR  
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR  
OTHER DEALINGS IN THE SOFTWARE.

## 1.142 oniguruma 6.9.10-r0

### 1.142.1 Available under license :

Oniguruma LICENSE

-----

Copyright (c) 2002-2021 K.Kosako <kkosako0@gmail.com>

All rights reserved.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions  
are met:

1. Redistributions of source code must retain the above copyright  
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright  
notice, this list of conditions and the following disclaimer in the  
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND  
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE  
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL  
DAMAGES (INCLUDING, BUT NOT LIMITED  
TO, PROCUREMENT OF SUBSTITUTE GOODS  
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT  
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY  
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF  
SUCH DAMAGE.

## 1.143 arptables 1.8.11

### 1.143.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA



Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free

program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause

any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete

machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the

Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

## How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute
it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if

necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into  
proprietary programs. If your program is a subroutine library, you may  
consider it more useful to permit linking proprietary applications with the  
library. If this  
is what you want to do, use the GNU Lesser General  
Public License instead of this License.

## 1.144 libedit 20250104.3.1-r1

### 1.144.1 Available under license :

Copyright (c) 1992, 1993

The Regents of the University of California. All rights reserved.

This code is derived from software contributed to Berkeley by  
Christos Zoulas of Cornell University.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions  
are met:

1. Redistributions of source code must retain the above copyright  
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright  
notice, this list of conditions and the following disclaimer in the  
documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors  
may be used to endorse or promote products derived from this software  
without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND  
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED  
TO, THE

IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE  
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL  
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS  
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT  
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY  
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF



SUCH DAMAGE.

# 1.145 gdbm 1.24-r0

## 1.145.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007, 2011 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2)

offer you this License  
giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS

### 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays

"Appropriate Legal Notices"

to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

#### 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other

than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

## 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

## 3. Protecting Users'

## Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

### 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

### 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7

additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

## 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This

alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge.

You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and

execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because

modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

## 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you



add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

## 8. Termination.

You

may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

#### 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

#### 10. Automatic Licensing of Downstream Recipients.

Each

time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

#### 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

## 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not

excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

#### 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

#### 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

#### 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

#### 16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### 17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief

idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
```

```
This program comes
```

```
with ABSOLUTELY NO WARRANTY; for details type `show w'.
```

```
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<http://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <<http://www.gnu.org/philosophy/why-not-lgpl.html>>.

## 1.146 skalibs 2.14.4.0-r0

## 1.146.1 Available under license :

### GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we



want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and

distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are

prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6.

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING

OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),  
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands

`show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

```
# Automatically generated by apkbuidl-cpan, template 4
# Contributor: Valery Kartel <valery.kartel@gmail.com>
# Maintainer: Celeste <cielesti@protonmail.com>
maintainer="Celeste <cielesti@protonmail.com>"
pkgname=perl-bud-resource
pkgver=1.2911
pkgrel=10
#_pkgreal is used by apkbuidl-cpan to find modules at MetaCpan
_pkgreal=BSD-Resource
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="https://metacpan.org/release/BSD-Resource/"
arch="all"
license="Artistic-2.0 OR LGPL-2.0"
depends="perl"
makedepends="perl-dev"
subpackages="$pkgname-doc"
source="https://cpan.metacpan.org/authors/id/J/JH/JHI/BSD-Resource-$pkgver.tar.gz"
buildid="$srcdir/$_pkgreal-$pkgver"
```

```
build() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    PERL_MM_USE_DEFAULT=1 perl -I. Makefile.PL \
    INSTALLDIRS=vendor \
    NO_PACKLIST=1 \
    NO_PERLLOCAL=1
    make
```

```

}

check() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    make test
}

package() {
    make DESTDIR="$pkgdir" install
}

sha512sums="
d0032d41c7c0468ed1c6d8f57b885f6cb97a5039d754c8cb60b2067daedaf53bd15fb6561a3d0f828df16dfa5417f663b
8065ba65f4fea16dc9262728b3b6b85
    BSD-Resource-1.2911.tar.gz
"

# Contributor: Maxim Karasev <begs@disroot.org>
# Maintainer: mio <miyopan@e.email>
pkgname=bsd-games
pkgver=3.3
pkgrel=2
pkgdesc="traditional text mode games from BSD"
url="https://bsd-games.sourceforge.io/"
arch="all"
license="BSD-3-Clause"
install="$pkgname.post-install"
makedepends="ncurses-dev coreutils" # configure causes busybox expr error
subpackages="$pkgname-doc"
source="https://sourceforge.net/projects/bsd-games/files/bsd-games-$pkgver.tar.gz
0001-bsd-games-scores-dir.patch
"

options="!check" # no tests

build() {
    # some GNU autoconf options are ignored, but it works
    ./configure \
        --build=$CBUILD \
        --host=$CHOST \
        --prefix=/usr \
        --sysconfdir=/etc \
        --mandir=/usr/share/man \
        --sourcedir=/usr/share/$pkgname
    make
}

package() {
    make DESTDIR="$pkgdir" install
    install -Dm644 LICENSE "$pkgdir"/usr/share/licenses/bsd-games/LICENSE

```



```
}
```

```
sha512sums="
aaf36d09d4fe68514a5c279063d7e77a9a9a84c447037cba974b29faa07eb80c2aab2379d0699c196b5c27ffe1b2bb2c6
8d3390143e0e62d5e3d210ccfb61294
bsd-games-3.3.tar.gz
e1def0d4835a877242c4173fae862e278a7858738e0bacacbb045a42bcabec2aa7c72d000bbd2421ce35fe66cdc1f52c0
577b7e278029ca5ed50f7aab22319f6 0001-bsd-games-scores-dir.patch
"
```

MIT License

Copyright (c) 2023 Leaf Corcoran

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```
# Maintainer: Natanael Copa <ncopa@alpinelinux.org>
```

```
pkgname=libart-lgpl
```

```
pkgver=2.3.21
```

```
pkgrel=8
```

```
pkgdesc="A library for high-performance 2D graphics"
```

```
url="https://www.levien.com/libart/"
```

```
arch="all"
```

```
options="!check" # No test suite.
```

```
license="LGPL-2.0-or-later"
```

```
subpackages="$pkgname-dev"
```

```
source="https://download.gnome.org/sources/libart-lgpl/2.3/libart-lgpl-$pkgver.tar.bz2"
```

```
"
```

```
builddir="$srcdir"/libart-lgpl-$pkgver
```

```
prepare() {
```

```
update_config_sub
```

```
default_prepare
}
```

```
build() {
./configure \
--build=$CBUILD \
--host=$CHOST \
--prefix=/usr
make
}
```

```
package() {
make DESTDIR="$pkgdir" install
}
```

```
sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020 libart_lgpl-2.3.21.tar.bz2"
mini_sendmail - accept email on behalf of real sendmail
```

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.  
All rights reserved.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions  
are met:

1. Redistributions of source code must retain the above copyright  
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright  
notice, this list of conditions and the following disclaimer in the  
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND  
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE  
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL  
DAMAGES (INCLUDING,  
BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS  
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT  
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY  
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF  
SUCH DAMAGE.

# Contributor: Carlo Landmeter <clandmeter@alpinelinux.org>

# Maintainer: Kevin Daudt <kdaudt@alpinelinux.org>

pkgname=spdx-licenses

pkgver=3.26.0

pkgrel=0

```

pkgdesc="Various data formats for the SPDX License List"
url="https://spdx.org/"
arch="noarch"
options="!check" # no test suite
license="CC-BY-3.0"
source="license-list-data-$pkgver.tar.gz::https://github.com/spdx/license-list-data/archive/v$pkgver.tar.gz"
builddir="$srcdir/license-list-data-$pkgver"
subpackages="$pkgname-list"

_types="html json rdfa rdft rdfturtle rdfoxml template text"

for type in $_types; do
    subpackages="$subpackages $pkgname-$type:_subpkg"
done

package() {
    mkdir -p "$pkgdir"
}

_subpkg() {
    local type=${subpkgname/$pkgname-/}
    pkgdesc="$pkgdesc ($type)"
    install_if="$pkgname=$pkgver-r$pkgrel"
    mkdir -p "$subpkgdir"/usr/share/spdx
    cp -r "$builddir"/$type "$subpkgdir"/usr/share/spdx/
}

list() {
    pkgdesc="$pkgdesc (licence list)"
    mkdir -p "$subpkgdir"/usr/share/spdx
    local i; for i in "$builddir"/text/*.txt;
    do
        local license=${i##*/}
        echo ${license%.*} >> "$subpkgdir"/usr/share/spdx/license.lst
    done
}

sha512sums="
67e618d4642dfe4f366935d9514d6f7941d3711e77c49bc1b340ac572d36015981b8f10c72a9f80f1d83e132ae7e5093c
0155e0c0e786e4777ee16d1cb7b885a license-list-data-3.26.0.tar.gz
"

# Contributor: Patrycja Rosa <alpine@ptrcnll.me>
# Maintainer: Patrycja Rosa <alpine@ptrcnll.me>
pkgname=bsd-compat-headers
pkgver=0.7.2
pkgrel=6
pkgdesc="BSD compatibility headers (cdefs, queue, tree)"
url="https://gitlab.alpinelinux.org/alpine/aports"

```

```

arch="noarch"
license="BSD-2-Clause AND BSD-3-Clause"
source="
cdefs.h
queue.h
tree.h
"
builddir="$srcdir"
options="!check" # just headers

package() {
    mkdir -p "$pkgdir"
    install -Dm644 -t "$pkgdir"/usr/include/sys \
        cdefs.h queue.h tree.h
}
sha512sums="
37c8fc73c7aea7b490f7850927e2bb91d12137c9e59e22c084146d515696dbc7973b5de92f4c987ba080dd2502ba8312
7006442c3f019b6447a620c0cae73178 cdefs.h
2f0d5e6e4dc3350285cf17009265dddcbe12431c111868eea39bc8cb038ab7c1f2acacbb21735c4e9d4a1fd106a8fc0f86
11ea33987d4faba37dde5ce6da0750 queue.h
d9ac210d81feb8ad2655bc80fb065d3fe20ae4417b32b4a1711e6738a4870140005c13373b5d1846ef3ce5ae6da45f2da
cef2092881eded0a2e94f6a07752ef3 tree.h
"

```

Original Copyright (c) 1985 by Supoj Sutanthavibul

Parts Copyright (c) 1989-2012 by Brian V. Smith

Parts Copyright (c) 1991 by Paul King

Other Copyrights may be found in various files

Main Xfig copyright notice:

Any party obtaining a copy of these files is granted, free of charge, a full and unrestricted irrevocable, world-wide, paid up, royalty-free, nonexclusive right and license to deal in this software and documentation files (the "Software"), including without limitation the rights to use, copy, modify, merge, publish and/or distribute copies of the Software, and to permit persons who receive copies from any such party to do so, with the only requirement being that this copyright notice remain intact.

No representations are made about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Parts Copyright (C) 1993 by Alan Richardson (mppa3@uk.ac.sussex.syma)

The text rotation code in w\_rottext.c was written by Alan Richardson.

The above copyright notice holds for this work as well.

Parts Copyright (c) 1994 by Anthony Dekker

The Kohonen neural network code for color optimization was written by Anthony Dekker.

Parts Copyright (c) 1995 by C. Blanc and C. Schlick

The X-Spline code was written Carole Blanc (blanc@labri.u-bordeaux.fr) and Christophe Schlick (schlick@labri.u-

bordeaux.fr) starting from an initial implementation done by C. Feuille, S. Grosbois, L. Maziere and L. Minihot as a student practice (Universite Bordeaux, France). For additional information about X-splines, see:

"X-Splines: A Spline Model Designed for the End User" by C. Blanc and C. Schlick, Proceedings of SIGGRAPH'95

<http://dept-info.labri.u-bordeaux.fr/~schlick/DOC/sig1.html>

# Contributor: Dhruvin Gandhi <contact@dhruvin>

# Maintainer: Patrycja Rosa <alpine@ptrcnull.me>

pkgname=py3-license-expression

\_pkgname=license\_expression

pkgver=30.4.1

pkgrel=0

pkgdesc="Library to parse, compare, simplify and normalize license expressions"

url="https://github.com/nexB/license-expression"

arch="noarch"

license="Apache-2.0"

depends="python3 py3-boolean.py"

makedepends="py3-setuptools py3-setuptools\_scm py3-gp517 py3-wheel"

checkdepends="py3-pytest py3-pytest-xdist"

subpackages="\$pkgname-pyc"

source="https://files.pythonhosted.org/packages/source/l/\_pkgname/\_pkgname-\$pkgver.tar.gz"

builddir="\$srcdir/\$\_pkgname-\$pkgver"

```
build() {  
    gp517 build-wheel \  
        --wheel-dir .dist \  
        --output-fd 3 3>&1 >&2  
}
```

```
check() {  
    python3 -m venv --clear --without-pip --system-site-packages .testenv  
    .testenv/bin/python3 -m installer .dist/*.whl  
    .testenv/bin/python3 -m pytest  
}
```

```
package() {  
    python3 -m installer -d "$pkgdir" \  
        .dist/*.whl  
}
```

```
sha512sums="  
114c630f098939757e49187e35b4e54883266a2ee9ca283724992420ef4c860f9606a8342da0ab74d93627c10805f312  
fda3f699cbc7c4fbd47a71e220440fb0  
    license_expression-30.4.1.tar.gz  
"
```

```
/* $NetBSD: queue.h,v 1.70 2015/11/02 15:21:23 christos Exp $ */
```

```
/*
```

```
* Copyright (c) 1991, 1993
```

```
* The Regents of the University of California. All rights reserved.
```

\*  
 \* Redistribution and use in source and binary forms, with or without  
 \* modification, are permitted provided that the following conditions  
 \* are met:  
 \* 1. Redistributions of source code must retain the above copyright  
 \* notice, this list of conditions and the following disclaimer.  
 \* 2. Redistributions in binary form must reproduce the above copyright  
 \* notice, this list of conditions and the following disclaimer in the  
 \* documentation and/or other materials provided with the distribution.  
 \* 3. Neither the name of the University nor the names of its contributors  
 \* may be used to endorse or promote products derived from this software  
 \* without specific prior written permission.  
 \*  
 \* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND  
 \* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,  
 \* BUT NOT LIMITED TO, THE  
 \* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
 \* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE  
 \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL  
 \* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS  
 \* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
 \* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT  
 \* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY  
 \* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF  
 \* SUCH DAMAGE.  
 \*  
 \* @(#)queue.h 8.5 (Berkeley) 8/20/94  
 \*/

```
#ifndef _SYS_QUEUE_H_
#define _SYS_QUEUE_H_
```

```
/*
 * This file defines five types of data structures: singly-linked lists,
 * lists, simple queues, tail queues, and circular queues.
 *
 * A singly-linked list is headed by a single forward pointer. The
 * elements are singly linked for minimum space and
 * pointer manipulation
 * overhead at the expense of O(n) removal for arbitrary elements. New
 * elements can be added to the list after an existing element or at the
 * head of the list. Elements being removed from the head of the list
 * should use the explicit macro for this purpose for optimum
 * efficiency. A singly-linked list may only be traversed in the forward
 * direction. Singly-linked lists are ideal for applications with large
 * datasets and few or no removals or for implementing a LIFO queue.
 *
 * A list is headed by a single forward pointer (or an array of forward
```

- \* pointers for a hash table header). The elements are doubly linked
- \* so that an arbitrary element can be removed without a need to
- \* traverse the list. New elements can be added to the list before
- \* or after an existing element or at the head of the list. A list
- \* may only be traversed in the forward direction.
- \*
- \* A simple queue is headed by a pair of pointers, one the head of the
- \* list and the
- other to the tail of the list. The elements are singly
- \* linked to save space, so elements can only be removed from the
- \* head of the list. New elements can be added to the list after
- \* an existing element, at the head of the list, or at the end of the
- \* list. A simple queue may only be traversed in the forward direction.
- \*
- \* A tail queue is headed by a pair of pointers, one to the head of the
- \* list and the other to the tail of the list. The elements are doubly
- \* linked so that an arbitrary element can be removed without a need to
- \* traverse the list. New elements can be added to the list before or
- \* after an existing element, at the head of the list, or at the end of
- \* the list. A tail queue may be traversed in either direction.
- \*
- \* A circle queue is headed by a pair of pointers, one to the head of the
- \* list and the other to the tail of the list. The elements are doubly
- \* linked so that an arbitrary element can be removed without a need to
- \* traverse the list. New elements
- can be added to the list before or after
- \* an existing element, at the head of the list, or at the end of the list.
- \* A circle queue may be traversed in either direction, but has a more
- \* complex end of list detection.
- \*
- \* For details on the use of these macros, see the queue(3) manual page.
- \*/

/\*

- \* Include the definition of NULL only on NetBSD because sys/null.h
- \* is not available elsewhere. This conditional makes the header
- \* portable and it can simply be dropped verbatim into any system.
- \* The caveat is that on other systems some other header
- \* must provide NULL before the macros can be used.
- \*/

```

#ifdef __NetBSD__
#include <sys/null.h>
#endif

#ifdef QUEUEDEBUG
# if defined(_KERNEL)
#  define QUEUEDEBUG_ABORT(...) panic(__VA_ARGS__)
# else

```

```

# include <err.h>
# define QUEUEDEBUG_ABORT(...) err(1, __VA_ARGS__)
# endif
#endif

/*
 * Singly-linked List definitions.
 */
#define SLIST_HEAD(name, type) \
struct name { \
    struct type *slh_first; /* first \
    element */ \
}

#define SLIST_HEAD_INITIALIZER(head) \
{ NULL }

#define SLIST_ENTRY(type) \
struct { \
    struct type *sle_next; /* next element */ \
}

/*
 * Singly-linked List access methods.
 */
#define SLIST_FIRST(head) ((head)->slh_first)
#define SLIST_END(head) NULL
#define SLIST_EMPTY(head) ((head)->slh_first == NULL)
#define SLIST_NEXT(elm, field) ((elm)->field.sle_next)

#define SLIST_FOREACH(var, head, field) \
for((var) = (head)->slh_first; \
    (var) != SLIST_END(head); \
    (var) = (var)->field.sle_next)

#define SLIST_FOREACH_SAFE(var, head, field, tvar) \
for ((var) = SLIST_FIRST((head)); \
    (var) != SLIST_END(head) && \
    ((tvar) = SLIST_NEXT((var), field), 1); \
    (var) = (tvar))

/*
 * Singly-linked List functions.
 */
#define SLIST_INIT(head) do { \
    (head)->slh_first = SLIST_END(head); \
} while (/*CONSTCOND*/0)

```



```
#define SLIST_INSERT_AFTER(slistelm, elm, field) do { \
    (elm)->field.sle_next = (slistelm)->field.sle_next; \
    (slistelm)->field.sle_next \
    = (elm); \
} while (/*CONSTCOND*/0)
```

```
#define SLIST_INSERT_HEAD(head, elm, field) do { \
    (elm)->field.sle_next = (head)->slh_first; \
    (head)->slh_first = (elm); \
} while (/*CONSTCOND*/0)
```

```
#define SLIST_REMOVE_AFTER(slistelm, field) do { \
    (slistelm)->field.sle_next = \
        SLIST_NEXT(SLIST_NEXT((slistelm), field), field); \
} while (/*CONSTCOND*/0)
```

```
#define SLIST_REMOVE_HEAD(head, field) do { \
    (head)->slh_first = (head)->slh_first->field.sle_next; \
} while (/*CONSTCOND*/0)
```

```
#define SLIST_REMOVE(head, elm, type, field) do { \
    if ((head)->slh_first == (elm)) { \
        SLIST_REMOVE_HEAD((head), field); \
    } \
    else { \
        struct type *curelm = (head)->slh_first; \
        while(curelm->field.sle_next != (elm)) \
            curelm = curelm->field.sle_next; \
        curelm->field.sle_next = \
            curelm->field.sle_next->field.sle_next; \
    } \
} while (/*CONSTCOND*/0)
```

```
/*
```

```
* List definitions.
```

```
*/
```

```
#define LIST_HEAD(name, type) \
struct name { \
    struct type *lh_first; /* first element */ \
}
```

```
#define LIST_HEAD_INITIALIZER(head) \
{ NULL }
```

```
#define LIST_ENTRY(type) \
struct { \
    struct type *le_next; /* next element */ \
}
```

```

struct type **le_prev; /* address of previous next element */\
}

/*
 * List access methods.
 */
#define LIST_FIRST(head) ((head)->lh_first)
#define LIST_END(head) NULL
#define LIST_EMPTY(head) ((head)->lh_first == LIST_END(head))
#define LIST_NEXT(elm, field) ((elm)->field.le_next)

#define LIST_FOREACH(var, head, field) \
for ((var) = ((head)->lh_first); \
     (var) != LIST_END(head); \
     (var) = ((var)->field.le_next))

#define LIST_FOREACH_SAFE(var, head, field, tvar) \
for ((var) = LIST_FIRST((head)); \
     (var) != LIST_END(head) && \
     ((tvar) = LIST_NEXT((var), field), 1); \
     (var) = (tvar))

#define LIST_MOVE(head1, head2) do { \
LIST_INIT((head2)); \
if (!LIST_EMPTY((head1))) \
{ \
(head2)->lh_first = (head1)->lh_first; \
LIST_INIT((head1)); \
} \
} while (/*CONSTCOND*/0)

/*
 * List functions.
 */
#ifdef QUEUEDEBDEBUG
#define QUEUEDEBDEBUG_LIST_INSERT_HEAD(head, elm, field) \
if ((head)->lh_first && \
     (head)->lh_first->field.le_prev != &(head)->lh_first) \
QUEUEDEBDEBUG_ABORT("LIST_INSERT_HEAD %p %s:%d", (head), \
                     __FILE__, __LINE__);
#define QUEUEDEBDEBUG_LIST_OP(elm, field) \
if ((elm)->field.le_next && \
     (elm)->field.le_next->field.le_prev != \
     &(elm)->field.le_next) \
QUEUEDEBDEBUG_ABORT("LIST_* forw %p %s:%d", (elm), \
                     __FILE__, __LINE__); \
if (*(elm)->field.le_prev != (elm)) \
QUEUEDEBDEBUG_ABORT("LIST_* back %p %s:%d", (elm), \

```

```

    __FILE__, __LINE__);
#define QUEUEDEBUG_LIST_POSTREMOVE(elm, field) \
    (elm)->field.le_next = (void *)1L; \
    (elm)->field.le_prev = (void *)1L;
#else
#define QUEUEDEBUG_LIST_INSERT_HEAD(head, elm, field)
#define QUEUEDEBUG_LIST_OP(elm,
    field)
#define QUEUEDEBUG_LIST_POSTREMOVE(elm, field)
#endif

#define LIST_INIT(head) do { \
    (head)->lh_first = LIST_END(head); \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_AFTER(listelm, elm, field) do { \
    QUEUEDEBUG_LIST_OP((listelm), field) \
    if (((elm)->field.le_next = (listelm)->field.le_next) != \
        LIST_END(head)) \
        (listelm)->field.le_next->field.le_prev = \
            &(elm)->field.le_next; \
    (listelm)->field.le_next = (elm); \
    (elm)->field.le_prev = &(listelm)->field.le_next; \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_BEFORE(listelm, elm, field) do { \
    QUEUEDEBUG_LIST_OP((listelm), field) \
    (elm)->field.le_prev = (listelm)->field.le_prev; \
    (elm)->field.le_next = (listelm); \
    *(listelm)->field.le_prev = (elm); \
    (listelm)->field.le_prev = &(elm)->field.le_next; \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_HEAD(head, elm, field) do { \
    QUEUEDEBUG_LIST_INSERT_HEAD((head), (elm), field) \
    if (((elm)->field.le_next \
        = (head)->lh_first) != LIST_END(head)) \
        (head)->lh_first->field.le_prev = &(elm)->field.le_next; \
    (head)->lh_first = (elm); \
    (elm)->field.le_prev = &(head)->lh_first; \
} while (/*CONSTCOND*/0)

#define LIST_REMOVE(elm, field) do { \
    QUEUEDEBUG_LIST_OP((elm), field) \
    if ((elm)->field.le_next != NULL) \
        (elm)->field.le_next->field.le_prev = \
            (elm)->field.le_prev; \
    *(elm)->field.le_prev = (elm)->field.le_next; \
} while (/*CONSTCOND*/0)

```

```

QUEUEDEBUG_LIST_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

#define LIST_REPLACE(elm, elm2, field) do { \
if (((elm2)->field.le_next = (elm)->field.le_next) != NULL) \
(elm2)->field.le_next->field.le_prev = \
    &(elm2)->field.le_next; \
(elm2)->field.le_prev = (elm)->field.le_prev; \
*(elm2)->field.le_prev = (elm2); \
QUEUEDEBUG_LIST_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

/*
 * Simple queue definitions.
 */
#define SIMPLEQ_HEAD(name, type) \
struct name { \
    struct \
    type *sqh_first; /* first element */ \
    struct type **sqh_last; /* addr of last next element */ \
}

#define SIMPLEQ_HEAD_INITIALIZER(head) \
{ NULL, &(head).sqh_first }

#define SIMPLEQ_ENTRY(type) \
struct { \
    struct type *sqe_next; /* next element */ \
}

/*
 * Simple queue access methods.
 */
#define SIMPLEQ_FIRST(head) ((head)->sqh_first)
#define SIMPLEQ_END(head) NULL
#define SIMPLEQ_EMPTY(head) ((head)->sqh_first == SIMPLEQ_END(head))
#define SIMPLEQ_NEXT(elm, field) ((elm)->field.sqe_next)

#define SIMPLEQ_FOREACH(var, head, field) \
for ((var) = ((head)->sqh_first); \
    (var) != SIMPLEQ_END(head); \
    (var) = ((var)->field.sqe_next))

#define SIMPLEQ_FOREACH_SAFE(var, head, field, next) \
for ((var) = ((head)->sqh_first); \
    (var) != SIMPLEQ_END(head) && \
    ((next = ((var)->field.sqe_next)), 1); \
    (var) = (next))

```

```

/*
 * Simple queue functions.
 */
#define SIMPLEQ_INIT(head) do { \
    (head)->sqh_first = NULL; \
    (head)->sqh_last \
    = &(head)->sqh_first; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_HEAD(head, elm, field) do { \
    if (((elm)->field.sqe_next = (head)->sqh_first) == NULL) \
        (head)->sqh_last = &(elm)->field.sqe_next; \
    (head)->sqh_first = (elm); \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_TAIL(head, elm, field) do { \
    (elm)->field.sqe_next = NULL; \
    *(head)->sqh_last = (elm); \
    (head)->sqh_last = &(elm)->field.sqe_next; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_AFTER(head, listelm, elm, field) do { \
    if (((elm)->field.sqe_next = (listelm)->field.sqe_next) == NULL) \
        (head)->sqh_last = &(elm)->field.sqe_next; \
    (listelm)->field.sqe_next = (elm); \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE_HEAD(head, field) do { \
    if (((head)->sqh_first = (head)->sqh_first->field.sqe_next) == NULL) \
        (head)->sqh_last = &(head)->sqh_first; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE_AFTER(head, elm, field) do { \
    if \
        (((elm)->field.sqe_next = (elm)->field.sqe_next->field.sqe_next) \
         == NULL) \
        (head)->sqh_last = &(elm)->field.sqe_next; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE(head, elm, type, field) do { \
    if ((head)->sqh_first == (elm)) { \
        SIMPLEQ_REMOVE_HEAD((head), field); \
    } else { \
        struct type *curelm = (head)->sqh_first; \
        while (curelm->field.sqe_next != (elm)) \
            curelm = curelm->field.sqe_next; \
        if ((curelm->field.sqe_next = \

```

```

    curelm->field.sqe_next->field.sqe_next) == NULL) \
    (head)->sqh_last = &(curelm)->field.sqe_next; \
} \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_CONCAT(head1, head2) do { \
    if (!SIMPLEQ_EMPTY((head2))) { \
        *(head1)->sqh_last = (head2)->sqh_first; \
        (head1)->sqh_last = (head2)->sqh_last; \
        SIMPLEQ_INIT((head2)); \
    } \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_LAST(head, type, field) \
(SIMPLEQ_EMPTY((head)) ? \
    NULL : \
    ((struct type \
    *) (void *) \
    ((char *)((head)->sqh_last) - offsetof(struct type, field))))

/*
 * Tail queue definitions.
 */

#define _TAILQ_HEAD(name, type, qual) \
struct name { \
    qual type *tqh_first; /* first element */ \
    qual type *qual *tqh_last; /* addr of last next element */ \
}

#define TAILQ_HEAD(name, type) _TAILQ_HEAD(name, struct type,)

#define TAILQ_HEAD_INITIALIZER(head) \
{ TAILQ_END(head), &(head).tqh_first }

#define _TAILQ_ENTRY(type, qual) \
struct { \
    qual type *tqe_next; /* next element */ \
    qual type *qual *tqe_prev; /* address of previous next element */ \
}

#define TAILQ_ENTRY(type) _TAILQ_ENTRY(struct type,)

/*
 * Tail queue access methods.
 */

#define TAILQ_FIRST(head) ((head)->tqh_first)
#define TAILQ_END(head) (NULL)
#define TAILQ_NEXT(elm, field) ((elm)->field.tqe_next)
#define TAILQ_LAST(head, headname) \
    (((struct headname *) (void *)((head)->tqh_last))->tqh_last)

```

```

#define TAILQ_PREV(elm, headname, field) \
    (((struct headname \
    *) (void *) ((elm)->field.tqe_prev))->tqh_last))
#define TAILQ_EMPTY(head) (TAILQ_FIRST(head) == TAILQ_END(head))

#define TAILQ_FOREACH(var, head, field) \
    for ((var) = ((head)->tqh_first); \
        (var) != TAILQ_END(head); \
        (var) = ((var)->field.tqe_next))

#define TAILQ_FOREACH_SAFE(var, head, field, next) \
    for ((var) = ((head)->tqh_first); \
        (var) != TAILQ_END(head) && \
        ((next) = TAILQ_NEXT(var, field), 1); (var) = (next))

#define TAILQ_FOREACH_REVERSE(var, head, headname, field) \
    for ((var) = TAILQ_LAST((head), headname); \
        (var) != TAILQ_END(head); \
        (var) = TAILQ_PREV((var), headname, field))

#define TAILQ_FOREACH_REVERSE_SAFE(var, head, headname, field, prev) \
    for ((var) = TAILQ_LAST((head), headname); \
        (var) != TAILQ_END(head) && \
        ((prev) = TAILQ_PREV((var), headname, field), 1); (var) = (prev))

/*
 * Tail queue functions.
 */
#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_TAILQ_INSERT_HEAD(head, \
    elm, field) \
    if ((head)->tqh_first && \
        (head)->tqh_first->field.tqe_prev != &(head)->tqh_first) \
        QUEUEDEBUG_ABORT("TAILQ_INSERT_HEAD %p %s:%d", (head), \
            __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_INSERT_TAIL(head, elm, field) \
    if (*(head)->tqh_last != NULL) \
        QUEUEDEBUG_ABORT("TAILQ_INSERT_TAIL %p %s:%d", (head), \
            __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_OP(elm, field) \
    if ((elm)->field.tqe_next && \
        (elm)->field.tqe_next->field.tqe_prev != \
        &(elm)->field.tqe_next) \
        QUEUEDEBUG_ABORT("TAILQ_* forw %p %s:%d", (elm), \
            __FILE__, __LINE__); \
    if (*(elm)->field.tqe_prev != (elm)) \
        QUEUEDEBUG_ABORT("TAILQ_* back %p %s:%d", (elm), \

```

```

    __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_PREREMOVE(head, elm, field) \
    if ((elm)->field.tqe_next == NULL && \
        (head)->tqh_last != &(elm)->field.tqe_next) \
        QUEUEDEBUG_ABORT("TAILQ_PREREMOVE head %p elm %p %s:%d", \
            (head), (elm), __FILE__, \
            __LINE__);
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field) \
    (elm)->field.tqe_next = (void *)1L; \
    (elm)->field.tqe_prev = (void *)1L;
#else
#define QUEUEDEBUG_TAILQ_INSERT_HEAD(head, elm, field)
#define QUEUEDEBUG_TAILQ_INSERT_TAIL(head, elm, field)
#define QUEUEDEBUG_TAILQ_OP(elm, field)
#define QUEUEDEBUG_TAILQ_PREREMOVE(head, elm, field)
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field)
#endif

#define TAILQ_INIT(head) do { \
    (head)->tqh_first = TAILQ_END(head); \
    (head)->tqh_last = &(head)->tqh_first; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_HEAD(head, elm, field) do { \
    QUEUEDEBUG_TAILQ_INSERT_HEAD((head), (elm), field) \
    if (((elm)->field.tqe_next = (head)->tqh_first) != TAILQ_END(head)) \
        (head)->tqh_first->field.tqe_prev = \
            &(elm)->field.tqe_next; \
    else \
        (head)->tqh_last = &(elm)->field.tqe_next; \
    (head)->tqh_first = (elm); \
    (elm)->field.tqe_prev = &(head)->tqh_first; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_TAIL(head, \
    elm, field) do { \
    QUEUEDEBUG_TAILQ_INSERT_TAIL((head), (elm), field) \
    (elm)->field.tqe_next = TAILQ_END(head); \
    (elm)->field.tqe_prev = (head)->tqh_last; \
    *(head)->tqh_last = (elm); \
    (head)->tqh_last = &(elm)->field.tqe_next; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_AFTER(head, listelm, elm, field) do { \
    QUEUEDEBUG_TAILQ_OP((listelm), field) \
    if (((elm)->field.tqe_next = (listelm)->field.tqe_next) != \
        TAILQ_END(head)) \
        (elm)->field.tqe_next->field.tqe_prev = \

```



```

    &(elm)->field.tqe_next; \
else \
    (head)->tqh_last = &(elm)->field.tqe_next; \
    (listelm)->field.tqe_next = (elm); \
    (elm)->field.tqe_prev = &(listelm)->field.tqe_next; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_BEFORE(listelm, elm, field) do { \
    QUEUEDEBDEBUG_TAILQ_OP((listelm), field) \
    (elm)->field.tqe_prev = (listelm)->field.tqe_prev; \
    (elm)->field.tqe_next = (listelm); \
    *(listelm)->field.tqe_prev = (elm); \
    (listelm)->field.tqe_prev \
    = &(elm)->field.tqe_next; \
} while (/*CONSTCOND*/0)

#define TAILQ_REMOVE(head, elm, field) do { \
    QUEUEDEBDEBUG_TAILQ_PREREMOVE((head), (elm), field) \
    QUEUEDEBDEBUG_TAILQ_OP((elm), field) \
    if (((elm)->field.tqe_next) != TAILQ_END(head)) \
        (elm)->field.tqe_next->field.tqe_prev = \
            (elm)->field.tqe_prev; \
    else \
        (head)->tqh_last = (elm)->field.tqe_prev; \
    *(elm)->field.tqe_prev = (elm)->field.tqe_next; \
    QUEUEDEBDEBUG_TAILQ_POSTREMOVE((elm), field); \
} while (/*CONSTCOND*/0)

#define TAILQ_REPLACE(head, elm, elm2, field) do { \
    if (((elm2)->field.tqe_next = (elm)->field.tqe_next) != \
        TAILQ_END(head)) \
        (elm2)->field.tqe_next->field.tqe_prev = \
            &(elm2)->field.tqe_next; \
    else \
        (head)->tqh_last = &(elm2)->field.tqe_next; \
    (elm2)->field.tqe_prev = (elm)->field.tqe_prev; \
    *(elm2)->field.tqe_prev = (elm2); \
    QUEUEDEBDEBUG_TAILQ_POSTREMOVE((elm), \
    field); \
} while (/*CONSTCOND*/0)

#define TAILQ_CONCAT(head1, head2, field) do { \
    if (!TAILQ_EMPTY(head2)) { \
        *(head1)->tqh_last = (head2)->tqh_first; \
        (head2)->tqh_first->field.tqe_prev = (head1)->tqh_last; \
        (head1)->tqh_last = (head2)->tqh_last; \
        TAILQ_INIT((head2)); \
    } \
}

```

```

} while (/*CONSTCOND*/0)

/*
 * Singly-linked Tail queue declarations.
 */
#define STAILQ_HEAD(name, type)  \
struct name {                    \
    struct type *stqh_first; /* first element */ \
    struct type **stqh_last; /* addr of last next element */ \
}

#define STAILQ_HEAD_INITIALIZER(head)  \
{ NULL, &(head).stqh_first }

#define STAILQ_ENTRY(type)  \
struct {                    \
    struct type *stqe_next; /* next element */ \
}

/*
 * Singly-linked Tail queue access methods.
 */
#define STAILQ_FIRST(head) ((head)->stqh_first)
#define STAILQ_END(head) NULL
#define STAILQ_NEXT(elm, field) ((elm)->field.stqe_next)
#define STAILQ_EMPTY(head) (STAILQ_FIRST(head) \
== STAILQ_END(head))

/*
 * Singly-linked Tail queue functions.
 */
#define STAILQ_INIT(head) do { \
    (head)->stqh_first = NULL; \
    (head)->stqh_last = &(head)->stqh_first; \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_HEAD(head, elm, field) do { \
    if (((elm)->field.stqe_next = (head)->stqh_first) == NULL) \
        (head)->stqh_last = &(elm)->field.stqe_next; \
    (head)->stqh_first = (elm); \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_TAIL(head, elm, field) do { \
    (elm)->field.stqe_next = NULL; \
    *(head)->stqh_last = (elm); \
    (head)->stqh_last = &(elm)->field.stqe_next; \
} while (/*CONSTCOND*/0)

```

```
#define STAILQ_INSERT_AFTER(head, listelm, elm, field) do { \
    if (((elm)->field.stqe_next = (listelm)->field.stqe_next) == NULL)\
        (head)->stqh_last = &(elm)->field.stqe_next; \
    (listelm)->field.stqe_next = (elm); \
} while (/*CONSTCOND*/0)
```

```
#define STAILQ_REMOVE_HEAD(head, field) do { \
    if (((head)->stqh_first = (head)->stqh_first->field.stqe_next) \
        == NULL) \
        (head)->stqh_last = &(head)->stqh_first; \
} while (/*CONSTCOND*/0)
```

```
#define STAILQ_REMOVE(head, elm, type, field) do { \
    if ((head)->stqh_first == (elm)) { \
        STAILQ_REMOVE_HEAD((head), field); \
    } else { \
        struct type *curelm = (head)->stqh_first; \
        while (curelm->field.stqe_next != (elm)) \
            curelm = curelm->field.stqe_next; \
        if ((curelm->field.stqe_next = \
            curelm->field.stqe_next->field.stqe_next) == NULL) \
            (head)->stqh_last = &(curelm)->field.stqe_next; \
    } \
} while (/*CONSTCOND*/0)
```

```
#define STAILQ_FOREACH(var, head, field) \
    for ((var) = ((head)->stqh_first); \
        (var); \
        (var) = ((var)->field.stqe_next))
```

```
#define STAILQ_FOREACH_SAFE(var, head, field, tvar) \
    for ((var) = STAILQ_FIRST((head)); \
        (var) && ((tvar) = STAILQ_NEXT((var), field), 1); \
        (var) = (tvar))
```

```
#define STAILQ_CONCAT(head1, head2) do { \
    if (!STAILQ_EMPTY((head2))) { \
        *(head1)->stqh_last \
        = (head2)->stqh_first; \
        (head1)->stqh_last = (head2)->stqh_last; \
        STAILQ_INIT((head2)); \
    } \
} while (/*CONSTCOND*/0)
```

```
#define STAILQ_LAST(head, type, field) \
    (STAILQ_EMPTY((head)) ? \
    NULL : \
    ((struct type *) (void *) \
        ((head)->stqh_last->field.stqe_next)))
```

```
((char *)((head)->stqh_last) - offsetof(struct type, field))))
```

```
#ifndef _KERNEL
```

```
/*
```

```
* Circular queue definitions. Do not use. We still keep the macros
```

```
* for compatibility but because of pointer aliasing issues their use
```

```
* is discouraged!
```

```
*/
```

```
/*
```

```
* __launder_type(): We use this ugly hack to work around the the compiler
```

```
* noticing that two types may not alias each other and elide tests in code.
```

```
* We hit this in the CIRCLEQ macros when comparing 'struct name *' and
```

```
* 'struct type *' (see CIRCLEQ_HEAD()). Modern compilers (such as GCC
```

```
* 4.8) declare these comparisons as always false, causing the code to
```

```
* not run as designed.
```

```
*
```

```
* This hack is only to be used for comparisons and thus can be fully const.
```

```
* Do not use for assignment.
```

```
*
```

```
* If we ever choose to change the ABI of the CIRCLEQ macros, we could fix
```

```
* this by changing the head/tail sentinel values, but see the note above
```

```
* this one.
```

```
*/
```

```
static __inline const void * __launder_type(const void *);
```

```
static __inline const void *
```

```
__launder_type(const void *__x)
```

```
{
```

```
__asm __volatile("" : "+r" (__x));
```

```
return __x;
```

```
}
```

```
#if defined(QUEUEDEBUG)
```

```
#define QUEUEDEBUG_CIRCLEQ_HEAD(head, field) \
```

```
if ((head)->cqh_first != CIRCLEQ_ENDC(head) && \
```

```
(head)->cqh_first->field.cqe_prev != CIRCLEQ_ENDC(head)) \
```

```
QUEUEDEBUG_ABORT("CIRCLEQ head forw %p %s:%d", (head), \
```

```
__FILE__, __LINE__); \
```

```
if ((head)->cqh_last != CIRCLEQ_ENDC(head) && \
```

```
(head)->cqh_last->field.cqe_next != CIRCLEQ_ENDC(head)) \
```

```
QUEUEDEBUG_ABORT("CIRCLEQ head back %p %s:%d", (head), \
```

```
__FILE__, __LINE__);
```

```
#define QUEUEDEBUG_CIRCLEQ_ELM(head, elm, field) \
```

```
if ((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) { \
```

```
if ((head)->cqh_last != (elm)) \
```

```
QUEUEDEBUG_ABORT("CIRCLEQ
```

```

elm last %p %s:%d", \
    (elm), __FILE__, __LINE__); \
} else { \
    if ((elm)->field.cqe_next->field.cqe_prev != (elm)) \
        QUEUEDEBUG_ABORT("CIRCLEQ elm forw %p %s:%d", \
            (elm), __FILE__, __LINE__); \
} \
if ((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) { \
    if ((head)->cqh_first != (elm)) \
        QUEUEDEBUG_ABORT("CIRCLEQ elm first %p %s:%d", \
            (elm), __FILE__, __LINE__); \
} else { \
    if ((elm)->field.cqe_prev->field.cqe_next != (elm)) \
        QUEUEDEBUG_ABORT("CIRCLEQ elm prev %p %s:%d", \
            (elm), __FILE__, __LINE__); \
}
#define QUEUEDEBUG_CIRCLEQ_POSTREMOVE(elm, field) \
    (elm)->field.cqe_next = (void *)1L; \
    (elm)->field.cqe_prev = (void *)1L;
#else
#define QUEUEDEBUG_CIRCLEQ_HEAD(head, field)
#define QUEUEDEBUG_CIRCLEQ_ELM(head, elm, field)
#define QUEUEDEBUG_CIRCLEQ_POSTREMOVE(elm, field)
#endif

#define CIRCLEQ_HEAD(name, type) \
struct name { \
    struct type *cqh_first; /* \
    first element */ \
    struct type *cqh_last; /* last element */ \
}

#define CIRCLEQ_HEAD_INITIALIZER(head) \
{ CIRCLEQ_END(&head), CIRCLEQ_END(&head) }

#define CIRCLEQ_ENTRY(type) \
struct { \
    struct type *cqe_next; /* next element */ \
    struct type *cqe_prev; /* previous element */ \
}

/*
 * Circular queue functions.
 */

#define CIRCLEQ_INIT(head) do { \
    (head)->cqh_first = CIRCLEQ_END(head); \
    (head)->cqh_last = CIRCLEQ_END(head); \
} while (/*CONSTCOND*/0)

```

```

#define CIRCLEQ_INSERT_AFTER(head, listelm, elm, field) do { \
    QUEUEDEBDEBUG_CIRCLEQ_HEAD((head), field) \
    QUEUEDEBDEBUG_CIRCLEQ_ELM((head), (listelm), field) \
    (elm)->field.cqe_next = (listelm)->field.cqe_next; \
    (elm)->field.cqe_prev = (listelm); \
    if ((listelm)->field.cqe_next == CIRCLEQ_ENDC(head)) \
        (head)->cqh_last = (elm); \
    else \
        (listelm)->field.cqe_next->field.cqe_prev = (elm); \
    (listelm)->field.cqe_next = (elm); \
} while (/*CONSTCOND*/0)

```

```

#define CIRCLEQ_INSERT_BEFORE(head,
listelm, elm, field) do { \
    QUEUEDEBDEBUG_CIRCLEQ_HEAD((head), field) \
    QUEUEDEBDEBUG_CIRCLEQ_ELM((head), (listelm), field) \
    (elm)->field.cqe_next = (listelm); \
    (elm)->field.cqe_prev = (listelm)->field.cqe_prev; \
    if ((listelm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \
        (head)->cqh_first = (elm); \
    else \
        (listelm)->field.cqe_prev->field.cqe_next = (elm); \
    (listelm)->field.cqe_prev = (elm); \
} while (/*CONSTCOND*/0)

```

```

#define CIRCLEQ_INSERT_HEAD(head, elm, field) do { \
    QUEUEDEBDEBUG_CIRCLEQ_HEAD((head), field) \
    (elm)->field.cqe_next = (head)->cqh_first; \
    (elm)->field.cqe_prev = CIRCLEQ_END(head); \
    if ((head)->cqh_last == CIRCLEQ_ENDC(head)) \
        (head)->cqh_last = (elm); \
    else \
        (head)->cqh_first->field.cqe_prev = (elm); \
    (head)->cqh_first = (elm); \
} while (/*CONSTCOND*/0)

```

```

#define CIRCLEQ_INSERT_TAIL(head, elm, field) do { \
    QUEUEDEBDEBUG_CIRCLEQ_HEAD((head), field) \
    (elm)->field.cqe_next
= CIRCLEQ_END(head); \
    (elm)->field.cqe_prev = (head)->cqh_last; \
    if ((head)->cqh_first == CIRCLEQ_ENDC(head)) \
        (head)->cqh_first = (elm); \
    else \
        (head)->cqh_last->field.cqe_next = (elm); \
    (head)->cqh_last = (elm); \
} while (/*CONSTCOND*/0)

```

```

#define CIRCLEQ_REMOVE(head, elm, field) do { \
    QUEUEDEBUG_CIRCLEQ_HEAD((head), field) \
    QUEUEDEBUG_CIRCLEQ_ELM((head), (elm), field) \
    if ((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) \
        (head)->cqh_last = (elm)->field.cqe_prev; \
    else \
        (elm)->field.cqe_next->field.cqe_prev = \
            (elm)->field.cqe_prev; \
    if ((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \
        (head)->cqh_first = (elm)->field.cqe_next; \
    else \
        (elm)->field.cqe_prev->field.cqe_next = \
            (elm)->field.cqe_next; \
    QUEUEDEBUG_CIRCLEQ_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_FOREACH(var, head, field) \
    for ((var) = ((head)->cqh_first); \
        (var) != CIRCLEQ_ENDC(head); \
        (var) \
        = ((var)->field.cqe_next))

#define CIRCLEQ_FOREACH_REVERSE(var, head, field) \
    for ((var) = ((head)->cqh_last); \
        (var) != CIRCLEQ_ENDC(head); \
        (var) = ((var)->field.cqe_prev))

/*
 * Circular queue access methods.
 */

#define CIRCLEQ_FIRST(head) ((head)->cqh_first)
#define CIRCLEQ_LAST(head) ((head)->cqh_last)
/* For comparisons */
#define CIRCLEQ_ENDC(head) (__launter_type(head))
/* For assignments */
#define CIRCLEQ_END(head) ((void *)(head))
#define CIRCLEQ_NEXT(elm, field) ((elm)->field.cqe_next)
#define CIRCLEQ_PREV(elm, field) ((elm)->field.cqe_prev)
#define CIRCLEQ_EMPTY(head) \
    (CIRCLEQ_FIRST(head) == CIRCLEQ_ENDC(head))

#define CIRCLEQ_LOOP_NEXT(head, elm, field) \
    (((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) \
     ? ((head)->cqh_first) \
     : (elm)->field.cqe_next)
#define CIRCLEQ_LOOP_PREV(head, elm, field) \
    (((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \

```

```

? ((head)->cqh_last) \
:
(elm->field.cqe_prev))
#endif /* !_KERNEL */

#endif /* !_SYS_QUEUE_H_ */
# Contributor: Fabian Affolter <fabian@affolter-engineering.ch>
# Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>
pkgname=py3-flake8-copyright
_pkgname=flake8-copyright
pkgver=0.2.4
pkgrel=3
pkgdesc="Extension for flake8 which checks for copyrights"
options="!check" # No testsuite
url="https://github.com/savoirfairelinux/flake8-copyright"
arch="noarch"
license="MIT"
depends="py3-flake8 py3-setuptools"
makedepends="py3-gpep517 py3-wheel"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/${_pkgname:0:1}/${_pkgname}/${_pkgname}-${pkgver}.tar.gz"
builddir="$srcdir"/$_pkgname-$pkgver

replaces="py-flake8-copyright" # Backwards compatibility
provides="py-flake8-copyright=$pkgver-r$pkgrel" # Backwards compatibility

build() {
    gpep517 build-wheel \
        --wheel-dir .dist \
        --output-fd 3 3>&1 >&2
}

package() {
    gpep517 install-wheel --destdir "$pkgdir" \
        .dist/*.whl
}

sha512sums="
a6ab47e1bb715618f075c51f398e18180404871b3b0faf1c2d30701d5203db6bc23771bf22ffb5bafcd8ee856b9b64237
b316bd503ae7dceed6ca284ccd5a74d
flake8-copyright-0.2.4.tar.gz
"

#ifndef _SYS_CDEFS_H_
#define _SYS_CDEFS_H_

#warning usage of non-standard #include <sys/cdefs.h> is deprecated

#undef __P

```



```

#undef __PMT

#define __P(args) args
#define __PMT(args) args

#define __CONCAT(x,y) x ## y
#define __STRING(x) #x

#ifdef __cplusplus
# define __BEGIN_DECLS extern "C" {
# define __END_DECLS }
#else
# define __BEGIN_DECLS
# define __END_DECLS
#endif

#if defined(__GNUC__) && !defined(__cplusplus)
# define __THROW __attribute__((__nothrow__))
# define __NTH(fct) __attribute__((__nothrow__)) fct
#else
# define __THROW
# define __NTH(fct) fct
#endif

#endif /* _SYS_CDEFS_H_ */
# Maintainer: fossdd <fossdd@pwned.life>
pkgname=cargo-license
pkgver=0.6.1
pkgrel=1
pkgdesc="Cargo subcommand to see license of dependencies"
url="https://github.com/onur/cargo-license"
arch="all"
license="MIT"
makedepends="cargo cargo-auditable"
source="$pkgname-$pkgver.tar.gz::https://crates.io/api/v1/crates/cargo-license/$pkgver/download"

prepare() {
    default_prepare

    cargo fetch --target="$CTARGET" --locked
}

build() {
    cargo auditable build --release --frozen
}

check() {
    cargo test --frozen

```

```

}

package() {
    install -Dm755 target/release/cargo-license -t "$pkgdir"/usr/bin
}

sha512sums="
36215145e79965be090ae2f5ce6f5eb208465eb67a94c26bed1fe70a131c46c6dc35695e30852759b7366883fbc3caf00
272d796ac634f0a5e1d49232b6b29d5  cargo-license-0.6.1.tar.gz
"

/* $NetBSD: tree.h,v 1.20 2013/09/14 13:20:45 joerg Exp $ */
/* $OpenBSD: tree.h,v 1.13 2011/07/09 00:19:45 pirofti Exp $ */
/* Modified by Void Linux. */
/*
* Copyright 2002 Niels Provos <provos@citi.umich.edu>
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE
* AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

#ifdef _SYS_TREE_H_
#define _SYS_TREE_H_

#ifdef __GNUC__
#define __GNUC_PREREQ__(x, y) \
    ((__GNUC__ == (x) && __GNUC_MINOR__ >= (y)) || \
     (__GNUC__ > (x)))
#else
#define __GNUC_PREREQ__(x, y) 0

```

```

#endif

#if __GNUC_PREREQ__(2, 7) || defined(__lint__)
#define _sys_tree_h_unused __attribute__((__unused__))
#else
#define _sys_tree_h_unused /* delete */
#endif

/*
 * This file defines data structures for different types of trees:
 * splay trees and red-black trees.
 *
 * A splay tree is a self-organizing
 * data structure. Every operation
 * on the tree causes a splay to happen. The splay moves the requested
 * node to the root of the tree and partly rebalances it.
 *
 * This has the benefit that request locality causes faster lookups as
 * the requested nodes move to the top of the tree. On the other hand,
 * every lookup causes memory writes.
 *
 * The Balance Theorem bounds the total access time for m operations
 * and n inserts on an initially empty tree as  $O((m + n)\lg n)$ . The
 * amortized cost for a sequence of m accesses to a splay tree is  $O(\lg n)$ ;
 *
 * A red-black tree is a binary search tree with the node color as an
 * extra attribute. It fulfills a set of conditions:
 * - every search path from the root to a leaf consists of the
 *   same number of black nodes,
 * - each red node (except for the root) has a black parent,
 * - each leaf node is black.
 *
 * Every operation on a red-black tree is bounded as  $O(\lg n)$ .
 * The maximum height of a red-black tree is  $2\lg(n+1)$ .
 */

#define SPLAY_HEAD(name, type) \
struct name { \
    struct type *sph_root; /* root of the tree */ \
}

#define SPLAY_INITIALIZER(root) \
{ NULL }

#define SPLAY_INIT(root) do { \
    (root)->sph_root = NULL; \
} while (/*CONSTCOND*/ 0)

```

```

#define SPLAY_ENTRY(type)    \
struct {                    \
    struct type *spe_left; /* left element */ \
    struct type *spe_right; /* right element */ \
}

#define SPLAY_LEFT(elm, field) (elm)->field.spe_left
#define SPLAY_RIGHT(elm, field) (elm)->field.spe_right
#define SPLAY_ROOT(head) (head)->sph_root
#define SPLAY_EMPTY(head) (SPLAY_ROOT(head) == NULL)

/* SPLAY_ROTATE_{LEFT,RIGHT} expect that tmp hold SPLAY_{RIGHT,LEFT} */
#define SPLAY_ROTATE_RIGHT(head, tmp, field) do { \
    SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(tmp, field); \
    SPLAY_RIGHT(tmp, field) = (head)->sph_root; \
    (head)->sph_root = tmp; \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ROTATE_LEFT(head, tmp, field) do { \
    SPLAY_RIGHT((head)->sph_root, \
    field) = SPLAY_LEFT(tmp, field); \
    SPLAY_LEFT(tmp, field) = (head)->sph_root; \
    (head)->sph_root = tmp; \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKLEFT(head, tmp, field) do { \
    SPLAY_LEFT(tmp, field) = (head)->sph_root; \
    tmp = (head)->sph_root; \
    (head)->sph_root = SPLAY_LEFT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKRIGHT(head, tmp, field) do { \
    SPLAY_RIGHT(tmp, field) = (head)->sph_root; \
    tmp = (head)->sph_root; \
    (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ASSEMBLE(head, node, left, right, field) do { \
    SPLAY_RIGHT(left, field) = SPLAY_LEFT((head)->sph_root, field); \
    SPLAY_LEFT(right, field) = SPLAY_RIGHT((head)->sph_root, field); \
    SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(node, field); \
    SPLAY_RIGHT((head)->sph_root, field) = SPLAY_LEFT(node, field); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */

#define

```

```

    SPLAY_PROTOTYPE(name, type, field, cmp) \
void name##_SPLAY(struct name *, struct type *); \
void name##_SPLAY_MINMAX(struct name *, int); \
struct type *name##_SPLAY_INSERT(struct name *, struct type *); \
struct type *name##_SPLAY_REMOVE(struct name *, struct type *); \
\
/* Finds the node with the same key as elm */ \
static __inline struct type * \
name##_SPLAY_FIND(struct name *head, struct type *elm) \
{ \
    if (SPLAY_EMPTY(head)) \
        return(NULL); \
    name##_SPLAY(head, elm); \
    if ((cmp)(elm, (head->sph_root) == 0) \
        return (head->sph_root); \
    return (NULL); \
} \
\
static __inline _sys_tree_h_unused struct type * \
name##_SPLAY_NEXT(struct name *head, struct type *elm) \
{ \
    name##_SPLAY(head, elm); \
    if (SPLAY_RIGHT(elm, field) != NULL) { \
        elm = SPLAY_RIGHT(elm, field); \
        while (SPLAY_LEFT(elm, field) != NULL) { \
            elm = SPLAY_LEFT(elm, field); \
        } \
    } \
    else \
        elm = NULL; \
    return (elm); \
} \
\
static _sys_tree_h_unused __inline struct type * \
name##_SPLAY_MIN_MAX(struct name *head, int val) \
{ \
    name##_SPLAY_MINMAX(head, val); \
    return (SPLAY_ROOT(head)); \
} \
\
/* Main splay operation.
 * Moves node close to the key of elm to top
 */
#define SPLAY_GENERATE(name, type, field, cmp) \
struct type * \
name##_SPLAY_INSERT(struct name *head, struct type *elm) \
{ \
    if (SPLAY_EMPTY(head)) { \

```

```

    SPLAY_LEFT(elm, field) = SPLAY_RIGHT(elm, field) = NULL; \
} else { \
    int __comp; \
    name##_SPLAY(head, elm); \
    __comp = (cmp)(elm, (head)->sph_root); \
    if(__comp < 0) { \
        SPLAY_LEFT(elm, field) = SPLAY_LEFT((head)->sph_root, field);\
        SPLAY_RIGHT(elm, field) = (head)->sph_root; \
        SPLAY_LEFT((head)->sph_root, field) = NULL; \
    } else if (__comp > 0) { \
        SPLAY_RIGHT(elm,
field) = SPLAY_RIGHT((head)->sph_root, field);\
        SPLAY_LEFT(elm, field) = (head)->sph_root; \
        SPLAY_RIGHT((head)->sph_root, field) = NULL; \
    } else \
        return ((head)->sph_root); \
} \
(head)->sph_root = (elm); \
return (NULL); \
} \
\
struct type * \
name##_SPLAY_REMOVE(struct name *head, struct type *elm) \
{ \
    struct type *__tmp; \
    if (SPLAY_EMPTY(head)) \
        return (NULL); \
    name##_SPLAY(head, elm); \
    if ((cmp)(elm, (head)->sph_root) == 0) { \
        if (SPLAY_LEFT((head)->sph_root, field) == NULL) { \
            (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field);\
        } else { \
            __tmp = SPLAY_RIGHT((head)->sph_root, field);\
            (head)->sph_root = SPLAY_LEFT((head)->sph_root, field);\
            name##_SPLAY(head, elm); \
            SPLAY_RIGHT((head)->sph_root, field) = __tmp; \
        } \
        return (elm); \
    } \
    return (NULL); \
} \
\
void \
name##_SPLAY(struct
name *head, struct type *elm) \
{ \
    struct type __node, *__left, *__right, *__tmp; \
    int __comp; \

```

```

\
SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
__left = __right = &__node; \
\
while ((__comp = (cmp)(elm, (head)->sph_root)) != 0) { \
if (__comp < 0) { \
__tmp = SPLAY_LEFT((head)->sph_root, field);\
if (__tmp == NULL) \
break; \
if ((cmp)(elm, __tmp) < 0){ \
SPLAY_ROTATE_RIGHT(head, __tmp, field);\
if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
break; \
} \
SPLAY_LINKLEFT(head, __right, field);\
} else if (__comp > 0) { \
__tmp = SPLAY_RIGHT((head)->sph_root, field);\
if (__tmp == NULL) \
break; \
if ((cmp)(elm, __tmp) > 0){ \
SPLAY_ROTATE_LEFT(head, __tmp, field);\
if (SPLAY_RIGHT((head)->sph_root, field) == NULL)\
break; \
} \
SPLAY_LINKRIGHT(head, __left, field);\
} \
} \
SPLAY_ASSEMBLE(head,
&__node, __left, __right, field);\
} \
\
/* Splay with either the minimum or the maximum element \
* Used to find minimum or maximum element in tree. \
*/ \
void name##_SPLAY_MINMAX(struct name *head, int __comp) \
{ \
struct type __node, *__left, *__right, *__tmp; \
\
SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
__left = __right = &__node; \
\
while (1) { \
if (__comp < 0) { \
__tmp = SPLAY_LEFT((head)->sph_root, field);\
if (__tmp == NULL) \
break; \
if (__comp < 0){ \
SPLAY_ROTATE_RIGHT(head, __tmp, field);\

```

```

    if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
        break; \
} \
SPLAY_LINKLEFT(head, __right, field); \
} else if (__comp > 0) { \
    __tmp = SPLAY_RIGHT((head)->sph_root, field);\
    if (__tmp == NULL) \
        break; \
    if (__comp > 0) { \
        SPLAY_ROTATE_LEFT(head, __tmp, field);\
        if (SPLAY_RIGHT((head)->sph_root,
field) == NULL)\
            break; \
    } \
    SPLAY_LINKRIGHT(head, __left, field); \
} \
} \
SPLAY_ASSEMBLE(head, &__node, __left, __right, field); \
}

#define SPLAY_NEGINF -1
#define SPLAY_INF 1

#define SPLAY_INSERT(name, x, y) name##_SPLAY_INSERT(x, y)
#define SPLAY_REMOVE(name, x, y) name##_SPLAY_REMOVE(x, y)
#define SPLAY_FIND(name, x, y) name##_SPLAY_FIND(x, y)
#define SPLAY_NEXT(name, x, y) name##_SPLAY_NEXT(x, y)
#define SPLAY_MIN(name, x) (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_NEGINF))
#define SPLAY_MAX(name, x) (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_INF))

#define SPLAY_FOREACH(x, name, head) \
    for ((x) = SPLAY_MIN(name, head); \
        (x) != NULL; \
        (x) = SPLAY_NEXT(name, head, x))

/* Macros that define a red-black tree */
#define RB_HEAD(name, type) \
struct name { \
    struct type *rbh_root; /* root of the tree */ \
}

#define RB_INITIALIZER(root) \
{ \
    NULL \
}

#define RB_INIT(root) do { \

```



```

(root)->rbh_root = NULL;  \
} while (/*CONSTCOND*/ 0)

#define RB_BLACK 0
#define RB_RED 1
#define RB_ENTRY(type)  \
struct {  \
    struct type *rbe_left; /* left element */ \
    struct type *rbe_right; /* right element */ \
    struct type *rbe_parent; /* parent element */ \
    int rbe_color; /* node color */ \
}

#define RB_LEFT(elm, field) (elm)->field.rbe_left
#define RB_RIGHT(elm, field) (elm)->field.rbe_right
#define RB_PARENT(elm, field) (elm)->field.rbe_parent
#define RB_COLOR(elm, field) (elm)->field.rbe_color
#define RB_ROOT(head) (head)->rbh_root
#define RB_EMPTY(head) (RB_ROOT(head) == NULL)

#define RB_SET(elm, parent, field) do {  \
    RB_PARENT(elm, field) = parent;  \
    RB_LEFT(elm, field) = RB_RIGHT(elm, field) = NULL; \
    RB_COLOR(elm, field) = RB_RED;  \
} while (/*CONSTCOND*/ 0)

#define RB_SET_BLACKRED(black, red, field) do {  \
    RB_COLOR(black, field) = RB_BLACK;  \
    RB_COLOR(red, field) \
    = RB_RED;  \
} while (/*CONSTCOND*/ 0)

#ifndef RB_AUGMENT
#define RB_AUGMENT(x) do {} while (/*CONSTCOND*/ 0)
#endif

#define RB_ROTATE_LEFT(head, elm, tmp, field) do {  \
    (tmp) = RB_RIGHT(elm, field);  \
    if ((RB_RIGHT(elm, field) = RB_LEFT(tmp, field)) != NULL) { \
        RB_PARENT(RB_LEFT(tmp, field), field) = (elm); \
    }  \
    RB_AUGMENT(elm);  \
    if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
        if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
            RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
        else  \
            RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
    } else  \

```

```

(head)->rbh_root = (tmp); \
RB_LEFT(tmp, field) = (elm); \
RB_PARENT(elm, field) = (tmp); \
RB_AUGMENT(tmp); \
if ((RB_PARENT(tmp, field))) \
    RB_AUGMENT(RB_PARENT(tmp, field)); \
} while (/*CONSTCOND*/ 0)

#define RB_ROTATE_RIGHT(head, elm, tmp, field) do { \
    (tmp) = RB_LEFT(elm, field); \
    if ((RB_LEFT(elm, field) = RB_RIGHT(tmp, field)) \
        != NULL) { \
        RB_PARENT(RB_RIGHT(tmp, field), field) = (elm); \
    } \
    RB_AUGMENT(elm); \
    if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
        if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
            RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
        else \
            RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
    } else \
        (head)->rbh_root = (tmp); \
    RB_RIGHT(tmp, field) = (elm); \
    RB_PARENT(elm, field) = (tmp); \
    RB_AUGMENT(tmp); \
    if ((RB_PARENT(tmp, field))) \
        RB_AUGMENT(RB_PARENT(tmp, field)); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */
#define RB_PROTOTYPE(name, type, field, cmp) \
    RB_PROTOTYPE_INTERNAL(name, type, field, cmp,)
#define RB_PROTOTYPE_STATIC(name, type, field, cmp) \
    RB_PROTOTYPE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_PROTOTYPE_INTERNAL(name, type, field, cmp, attr) \
    attr void name##_RB_INSERT_COLOR(struct name *, struct \
        type *); \
    attr void name##_RB_REMOVE_COLOR(struct name *, struct type *, struct type *); \
    attr struct type *name##_RB_REMOVE(struct name *, struct type *); \
    attr struct type *name##_RB_INSERT(struct name *, struct type *); \
    attr struct type *name##_RB_FIND(struct name *, struct type *); \
    attr struct type *name##_RB_NFIND(struct name *, struct type *); \
    attr struct type *name##_RB_NEXT(struct type *); \
    attr struct type *name##_RB_PREV(struct type *); \
    attr struct type *name##_RB_MINMAX(struct name *, int); \
    \

/* Main rb operation.

```

```

* Moves node close to the key of elm to top
*/
#define RB_GENERATE(name, type, field, cmp) \
    RB_GENERATE_INTERNAL(name, type, field, cmp,)
#define RB_GENERATE_STATIC(name, type, field, cmp) \
    RB_GENERATE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_GENERATE_INTERNAL(name, type, field, cmp, attr) \
    attr void \
name##_RB_INSERT_COLOR(struct name *head, struct type *elm) \
{ \
    struct type \
    *parent, *gparent, *tmp; \
    while ((parent = RB_PARENT(elm, field)) != NULL && \
        RB_COLOR(parent, field) == RB_RED) { \
        gparent = RB_PARENT(parent, field); \
        if (parent == RB_LEFT(gparent, field)) { \
            tmp = RB_RIGHT(gparent, field); \
            if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
                RB_COLOR(tmp, field) = RB_BLACK; \
                RB_SET_BLACKRED(parent, gparent, field);\
                elm = gparent; \
                continue; \
            } \
            if (RB_RIGHT(parent, field) == elm) { \
                RB_ROTATE_LEFT(head, parent, tmp, field);\
                tmp = parent; \
                parent = elm; \
                elm = tmp; \
            } \
            RB_SET_BLACKRED(parent, gparent, field); \
            RB_ROTATE_RIGHT(head, gparent, tmp, field); \
        } else { \
            tmp = RB_LEFT(gparent, field); \
            if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
                RB_COLOR(tmp, field) = RB_BLACK; \
                RB_SET_BLACKRED(parent, gparent, field);\
                elm = gparent; \
                continue; \
            } \
            if (RB_LEFT(parent, field) == elm) \
{ \
                RB_ROTATE_RIGHT(head, parent, tmp, field);\
                tmp = parent; \
                parent = elm; \
                elm = tmp; \
            } \
            RB_SET_BLACKRED(parent, gparent, field); \
            RB_ROTATE_LEFT(head, gparent, tmp, field); \

```

```

    } \
} \
RB_COLOR(head->rbh_root, field) = RB_BLACK; \
} \
    \
attr void \
name##_RB_REMOVE_COLOR(struct name *head, struct type *parent, struct type *elm) \
{ \
    struct type *tmp; \
    while ((elm == NULL || RB_COLOR(elm, field) == RB_BLACK) && \
        elm != RB_ROOT(head)) { \
        if (RB_LEFT(parent, field) == elm) { \
            tmp = RB_RIGHT(parent, field); \
            if (RB_COLOR(tmp, field) == RB_RED) { \
                RB_SET_BLACKRED(tmp, parent, field); \
                RB_ROTATE_LEFT(head, parent, tmp, field);\
                tmp = RB_RIGHT(parent, field); \
            } \
            if ((RB_LEFT(tmp, field) == NULL || \
                RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) && \
                (RB_RIGHT(tmp, field) == NULL || \
                RB_COLOR(RB_RIGHT(tmp, \
field), field) == RB_BLACK)) {\
                RB_COLOR(tmp, field) = RB_RED; \
                elm = parent; \
                parent = RB_PARENT(elm, field); \
            } else { \
                if (RB_RIGHT(tmp, field) == NULL || \
                    RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK) {\
                    struct type *oleft; \
                    if ((oleft = RB_LEFT(tmp, field)) \
                        != NULL) \
                        RB_COLOR(oleft, field) = RB_BLACK;\
                    RB_COLOR(tmp, field) = RB_RED; \
                    RB_ROTATE_RIGHT(head, tmp, oleft, field);\
                    tmp = RB_RIGHT(parent, field); \
                } \
                RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
                RB_COLOR(parent, field) = RB_BLACK; \
                if (RB_RIGHT(tmp, field)) \
                    RB_COLOR(RB_RIGHT(tmp, field), field) = RB_BLACK;\
                RB_ROTATE_LEFT(head, parent, tmp, field);\
                elm = RB_ROOT(head); \
                break; \
            } \
        } else { \
            tmp = RB_LEFT(parent, field); \
            if (RB_COLOR(tmp, field) == RB_RED) { \

```

```

    RB_SET_BLACKRED(tmp, parent, field); \
    RB_ROTATE_RIGHT(head, parent, tmp, field);\
    tmp
= RB_LEFT(parent, field); \
} \
if ((RB_LEFT(tmp, field) == NULL || \
    RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) &&\
    (RB_RIGHT(tmp, field) == NULL || \
    RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK)) {\
    RB_COLOR(tmp, field) = RB_RED; \
    elm = parent; \
    parent = RB_PARENT(elm, field); \
} else { \
    if (RB_LEFT(tmp, field) == NULL || \
        RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) {\
        struct type *oright; \
        if ((oright = RB_RIGHT(tmp, field)) \
            != NULL) \
            RB_COLOR(oright, field) = RB_BLACK;\
        RB_COLOR(tmp, field) = RB_RED; \
        RB_ROTATE_LEFT(head, tmp, oright, field);\
        tmp = RB_LEFT(parent, field); \
    } \
    RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
    RB_COLOR(parent, field) = RB_BLACK; \
    if (RB_LEFT(tmp, field)) \
        RB_COLOR(RB_LEFT(tmp, field), field) = RB_BLACK;\
    RB_ROTATE_RIGHT(head, parent, tmp, field);\
    elm = RB_ROOT(head); \
    break; \
} \
} \
} \
if
(elm) \
    RB_COLOR(elm, field) = RB_BLACK; \
} \
\
attr struct type * \
name##_RB_REMOVE(struct name *head, struct type *elm) \
{ \
    struct type *child, *parent, *old = elm; \
    int color; \
    if (RB_LEFT(elm, field) == NULL) \
        child = RB_RIGHT(elm, field); \
    else if (RB_RIGHT(elm, field) == NULL) \
        child = RB_LEFT(elm, field); \
    else { \

```

```

struct type *left;  \
elm = RB_RIGHT(elm, field);  \
while ((left = RB_LEFT(elm, field)) != NULL) \
    elm = left;  \
child = RB_RIGHT(elm, field);  \
parent = RB_PARENT(elm, field);  \
color = RB_COLOR(elm, field);  \
if (child)  \
    RB_PARENT(child, field) = parent; \
if (parent) {  \
    if (RB_LEFT(parent, field) == elm) \
        RB_LEFT(parent, field) = child; \
    else  \
        RB_RIGHT(parent, field) = child; \
    RB_AUGMENT(parent);  \
} else  \
    RB_ROOT(head)
= child;  \
if (RB_PARENT(elm, field) == old) \
    parent = elm;  \
(elm)->field = (old)->field;  \
if (RB_PARENT(old, field)) {  \
    if (RB_LEFT(RB_PARENT(old, field), field) == old)\
        RB_LEFT(RB_PARENT(old, field), field) = elm;\
    else  \
        RB_RIGHT(RB_PARENT(old, field), field) = elm;\
    RB_AUGMENT(RB_PARENT(old, field)); \
} else  \
    RB_ROOT(head) = elm;  \
RB_PARENT(RB_LEFT(old, field), field) = elm; \
if (RB_RIGHT(old, field))  \
    RB_PARENT(RB_RIGHT(old, field), field) = elm; \
if (parent) {  \
    left = parent;  \
    do {  \
        RB_AUGMENT(left); \
    } while ((left = RB_PARENT(left, field)) != NULL); \
}  \
goto color;  \
}  \
parent = RB_PARENT(elm, field);  \
color = RB_COLOR(elm, field);  \
if (child)  \
    RB_PARENT(child, field) = parent; \
if (parent) {  \
    if (RB_LEFT(parent, field) == elm) \
        RB_LEFT(parent, field) = child; \
    else  \

```

```

    RB_RIGHT(parent,
field) = child; \
    RB_AUGMENT(parent); \
} else \
    RB_ROOT(head) = child; \
color: \
if (color == RB_BLACK) \
    name##_RB_REMOVE_COLOR(head, parent, child); \
return (old); \
} \
\
/* Inserts a node into the RB tree */ \
attr struct type * \
name##_RB_INSERT(struct name *head, struct type *elm) \
{ \
    struct type *tmp; \
    struct type *parent = NULL; \
    int comp = 0; \
    tmp = RB_ROOT(head); \
    while (tmp) { \
        parent = tmp; \
        comp = (cmp)(elm, parent); \
        if (comp < 0) \
            tmp = RB_LEFT(tmp, field); \
        else if (comp > 0) \
            tmp = RB_RIGHT(tmp, field); \
        else \
            return (tmp); \
    } \
    RB_SET(elm, parent, field); \
    if (parent != NULL) { \
        if (comp < 0) \
            RB_LEFT(parent, field) = elm; \
        else \
            RB_RIGHT(parent, field) = elm; \
        RB_AUGMENT(parent); \
    } else \
        RB_ROOT(head) = elm; \
    name##_RB_INSERT_COLOR(head,
elm); \
    return (NULL); \
} \
\
/* Finds the node with the same key as elm */ \
attr struct type * \
name##_RB_FIND(struct name *head, struct type *elm) \
{ \
    struct type *tmp = RB_ROOT(head); \

```

```

int comp;    \
while (tmp) {    \
    comp = cmp(elm, tmp);    \
    if (comp < 0)    \
        tmp = RB_LEFT(tmp, field);    \
    else if (comp > 0)    \
        tmp = RB_RIGHT(tmp, field);    \
    else    \
        return (tmp);    \
}    \
return (NULL);    \
}    \
    \
/* Finds the first node greater than or equal to the search key */\
attr struct type *    \
name##_RB_NFIND(struct name *head, struct type *elm)    \
{    \
    struct type *tmp = RB_ROOT(head);    \
    struct type *res = NULL;    \
    int comp;    \
    while (tmp) {    \
        comp = cmp(elm, tmp);    \
        if (comp < 0) {    \
            res = tmp;    \
            tmp = RB_LEFT(tmp, field);    \
        }    \
        else if (comp > 0)    \
            tmp =    \
RB_RIGHT(tmp, field);    \
        else    \
            return (tmp);    \
    }    \
    return (res);    \
}    \
    \
/* ARGSUSED */    \
attr struct type *    \
name##_RB_NEXT(struct type *elm)    \
{    \
    if (RB_RIGHT(elm, field)) {    \
        elm = RB_RIGHT(elm, field);    \
        while (RB_LEFT(elm, field))    \
            elm = RB_LEFT(elm, field);    \
    } else {    \
        if (RB_PARENT(elm, field) &&    \
            (elm == RB_LEFT(RB_PARENT(elm, field), field)))    \
            elm = RB_PARENT(elm, field);    \
        else {    \

```



```

while (RB_PARENT(elm, field) && \
      (elm == RB_RIGHT(RB_PARENT(elm, field), field)))\
    elm = RB_PARENT(elm, field); \
elm = RB_PARENT(elm, field); \
} \
} \
return (elm); \
} \
\
/* ARGSUSED */ \
attr struct type * \
name##_RB_PREV(struct type *elm) \
{ \
    if (RB_LEFT(elm, field)) { \
        elm = RB_LEFT(elm, field); \
        while (RB_RIGHT(elm, field)) \
            elm = RB_RIGHT(elm, field); \
    } \
    else { \
        if (RB_PARENT(elm, field) && \
            (elm == RB_RIGHT(RB_PARENT(elm, field), field))) \
            elm = RB_PARENT(elm, field); \
        else { \
            while (RB_PARENT(elm, field) && \
                  (elm == RB_LEFT(RB_PARENT(elm, field), field)))\
                elm = RB_PARENT(elm, field); \
            elm = RB_PARENT(elm, field); \
        } \
    } \
    return (elm); \
} \
\
attr struct type * \
name##_RB_MINMAX(struct name *head, int val) \
{ \
    struct type *tmp = RB_ROOT(head); \
    struct type *parent = NULL; \
    while (tmp) { \
        parent = tmp; \
        if (val < 0) \
            tmp = RB_LEFT(tmp, field); \
        else \
            tmp = RB_RIGHT(tmp, field); \
    } \
    return (parent); \
}

#define RB_NEGINF -1

```

```

#define RB_INF 1

#define RB_INSERT(name, x, y) name##_RB_INSERT(x, y)
#define RB_REMOVE(name, x, y) name##_RB_REMOVE(x, y)
#define RB_FIND(name, x, y) name##_RB_FIND(x, y)
#define RB_NFIND(name, x, y) name##_RB_NFIND(x,
y)
#define RB_NEXT(name, x, y) name##_RB_NEXT(y)
#define RB_PREV(name, x, y) name##_RB_PREV(y)
#define RB_MIN(name, x) name##_RB_MINMAX(x, RB_NEGINF)
#define RB_MAX(name, x) name##_RB_MINMAX(x, RB_INF)

#define RB_FOREACH(x, name, head) \
for ((x) = RB_MIN(name, head); \
(x) != NULL; \
(x) = name##_RB_NEXT(x))

#define RB_FOREACH_FROM(x, name, y) \
for ((x) = (y); \
((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
(x) = (y))

#define RB_FOREACH_SAFE(x, name, head, y) \
for ((x) = RB_MIN(name, head); \
((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
(x) = (y))

#define RB_FOREACH_REVERSE(x, name, head) \
for ((x) = RB_MAX(name, head); \
(x) != NULL; \
(x) = name##_RB_PREV(x))

#define RB_FOREACH_REVERSE_FROM(x, name, y) \
for ((x) = (y); \
((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
(x) = (y))

#define RB_FOREACH_REVERSE_SAFE(x, name,
head, y) \
for ((x) = RB_MAX(name, head); \
((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
(x) = (y))

#endif /* _SYS_TREE_H_ */
pkgname=linux-headers
pkgver=6.14.2 # Follow the latest Linux stable
_kernver=${pkgver%.*}
pkgrel=0

```

```
pkgdesc="Linux system headers"
url="https://kernel.org/"
arch="all" # Headers are architecture specific
license="GPL-2.0-only"
makedepends="perl"
options="!check !dbg !strip !tracedeps !archcheck"
source="https://kernel.org/pub/linux/kernel/v6.x/linux-${_kernver}.tar.xz
revert-broken-uapi.patch
0003-remove-inclusion-of-sysinfo.h-in-kernel.h.patch
```

# 1.147 wget 1.25.0-r1

## 1.147.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007, 2023 Free Software Foundation, Inc. <<https://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS

### 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

## 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major

Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

## 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction

and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

### 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

### 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

### 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is

released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

## 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the



product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source

in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

## 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty  
or limiting liability differently from the  
terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or  
author attributions in that material or in the Appropriate Legal  
Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or  
requiring that modified versions of such material be marked in  
reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or  
authors of the material; or
- e) Declining to grant rights under trademark law for use of some  
trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that  
material by anyone who conveys the material (or modified versions of  
it) with contractual assumptions of liability to the recipient, for  
any liability that these contractual assumptions directly impose  
on  
those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the

additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

#### 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

#### 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do

so.

#### 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

#### 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and

propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

#### 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

#### 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

#### 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's

public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

#### 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

#### 16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### 17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.



To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>  
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<https://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read

# 1.148 libeconf 0.6.3-r0

## 1.148.1 Available under license :

Copyright (c) 2019 SUSE LLC

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.149 tmux 3.5a-r1

## 1.149.1 Available under license :

THIS IS FOR INFORMATION ONLY, CODE IS UNDER THE LICENCE AT THE TOP OF ITS FILE.

The README, CHANGES, FAQ and TODO files are licensed under the ISC license. All other files have a license and copyright notice at their start, typically:

Copyright (c) <author>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF MIND, USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING

OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

# 1.150 fstrm 0.6.1-r4

## 1.150.1 Available under license :

Copyright (c) 2012-2014 by Farsight Security, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.  
Copyright (c) 2014 by Farsight Security, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.151 keyutils 1.6.3-r4

## 1.151.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. <<http://fsf.org/>>

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and

you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program  
(or a work based on it,  
under Section 2) in object code or executable form under the terms of  
Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program  
in object code or executable form with such  
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled  
to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute



software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies  
a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.

The hypothetical commands `show  
w' and `show c' should show the appropriate  
parts of the General Public License. Of course, the commands you use may  
be called something other than `show w' and `show c'; they could even be  
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your  
school, if any, to sign a "copyright disclaimer" for the program, if  
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into  
proprietary programs. If your program is a subroutine library, you may  
consider it more useful to permit linking proprietary applications with the  
library. If this is what you want to do, use the GNU Library General  
Public License instead of this License.

GNU LESSER GENERAL PUBLIC LICENSE  
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc. <<http://fsf.org/>>  
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts  
as the successor of the GNU Library Public License, version 2, hence  
the version number 2.1.]

#### Preamble

The licenses for most software are designed to take away your  
freedom to share and change it. By contrast, the GNU General Public  
Licenses are intended to guarantee your freedom to share and change  
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some  
specially designated software packages--typically libraries--of the  
Free Software Foundation and other authors who decide to use it. You

can use

it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser

General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL  
PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy,

and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore



covered by this License.

Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6.

Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application

to use the modified definitions.)

- b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library

facilities. This must  
be distributed under the terms of the  
Sections above.

b) Give prominent notice with the combined library of the fact  
that part of it is a work based on the Library, and explaining  
where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute  
the Library except as expressly provided under this License. Any  
attempt otherwise to copy, modify, sublicense, link with, or  
distribute the Library is void, and will automatically terminate your  
rights under this License. However, parties who have received copies,  
or rights, from you under this License will not have their licenses  
terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not  
signed it. However, nothing else grants you permission to modify or  
distribute the Library or its derivative works. These actions are  
prohibited by law if you do not accept this License. Therefore,  
by  
modifying or distributing the Library (or any work based on the  
Library), you indicate your acceptance of this License to do so, and  
all its terms and conditions for copying, distributing or modifying  
the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the  
Library), the recipient automatically receives a license from the  
original licensor to copy, distribute, link with or modify the Library  
subject to these terms and conditions. You may not impose any further  
restrictions on the recipients' exercise of the rights granted herein.  
You are not responsible for enforcing compliance by third parties with  
this License.

11. If, as a consequence of a court judgment or allegation of patent  
infringement or for any other reason (not limited to patent issues),  
conditions are imposed on you (whether by court order, agreement or  
otherwise) that contradict the conditions of this License, they do not  
excuse you from the conditions of this License.

If you cannot  
distribute so as to satisfy simultaneously your obligations under this  
License and any other pertinent obligations, then as a consequence you  
may not distribute the Library at all. For example, if a patent  
license would not permit royalty-free redistribution of the Library by  
all those who receive copies directly or indirectly through you, then  
the only way you could satisfy both it and this License would be to  
refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices.

Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser

General Public License from time to time.

Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving

the free status  
of all derivatives of our free software and of promoting the sharing  
and reuse of software generally.

## NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or

modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!

## 1.152 openssl 10.0\_p1-r9

### 1.152.1 Notifications :

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)

This product includes cryptographic software written by Eric Young ([eay@cryptsoft.com](mailto:eay@cryptsoft.com)).

This product includes software written by Tim Hudson ([tjh@cryptsoft.com](mailto:tjh@cryptsoft.com)).

### 1.152.2 Available under license :

/\*

\* Copyright (c) 2000 Ben Lindstrom. All rights reserved.

\*

\* Redistribution and use in source and binary forms, with or without  
\* modification, are permitted provided that the following conditions  
\* are met:

\* 1. Redistributions of source code must retain the above copyright  
\* notice, this list of conditions and the following disclaimer.

\* 2. Redistributions in binary form must reproduce the above copyright  
\* notice, this list of conditions and the following disclaimer in the  
\* documentation and/or other materials provided with the distribution.

```

*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
*/

```

```

#ifndef _BSD_WAITPID_H
#define _BSD_WAITPID_H

```

```

#ifndef HAVE_WAITPID
/* Clean out any potential issues */
#undef WIFEXITED
#undef WIFSTOPPED
#undef WIFSIGNALED

```

```

/* Define required functions to mimic a POSIX look and feel */
#define _W_INT(w) (*(int*)&(w)) /* convert union wait to int */
#define WIFEXITED(w) (!((_W_INT(w)) & 0377))
#define WIFSTOPPED(w) ((_W_INT(w)) & 0100)
#define WIFSIGNALED(w) (!WIFEXITED(w) && !WIFSTOPPED(w))
#define WEXITSTATUS(w) (int)(WIFEXITED(w) ? ((_W_INT(w) >> 8) & 0377) : -1)
#define WTERMSIG(w) (int)(WIFSIGNALED(w) ? (_W_INT(w) & 0177) : -1)
#define WCOREFLAG 0x80
#define WCOREDUMP(w) ((_W_INT(w)) & WCOREFLAG)

```

```

/* Prototype */
pid_t waitpid(int,
int *, int);

```

```

#endif /* !HAVE_WAITPID */
#endif /* _BSD_WAITPID_H */
/* $OpenBSD: poll.h,v 1.11 2003/12/10 23:10:08 millert Exp $ */

```

```

/*
* Copyright (c) 1996 Theo de Raadt
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions

```

\* are met:

- \* 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \*

\* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

\*/

/\* OPENBSD ORIGINAL: sys/sys/poll.h \*/

```
#ifndef _COMPAT_POLL_H_
#define _COMPAT_POLL_H_
```

```
#include <sys/types.h>
#ifdef HAVE_POLL_H
# include <poll.h>
#elif HAVE_SYS_POLL_H
# include <sys/poll.h>
#endif
```

```
#ifndef HAVE_STRUCT_POLLFD_FD
typedef struct pollfd {
    int fd;
    short events;
    short revents;
} pollfd_t;
```

```
#ifndef POLLIN
# define POLLIN 0x0001
#endif
#ifndef POLLPRI
# define POLLPRI 0x0002
#endif
#ifndef POLLOUT
# define POLLOUT 0x0004
#endif
```



```

#ifndef POLLERR
# define POLLERR 0x0008
#endif

#ifndef POLLHUP
# define POLLHUP 0x0010
#endif

#ifndef POLLNVAL
# define POLLNVAL 0x0020
#endif

#if 0
/* the following are
   currently not implemented */
#define POLLRDNORM 0x0040
#define POLLNORM POLLRDNORM
#define POLLWRNORM    POLLOUT
#define POLLRDBAND 0x0080
#define POLLWRBAND 0x0100
#endif

#define INFTIM (-1) /* not standard */
#endif /* !HAVE_STRUCT_POLLFD_FD */

#ifndef HAVE_NFDS_T
typedef unsigned int nfds_t;
#endif

#ifndef HAVE_POLL
int poll(struct pollfd *, nfds_t, int);
#endif

#ifndef HAVE_PPOLL
int ppoll(struct pollfd *, nfds_t, const struct timespec *, const sigset_t *);
#endif

#endif /* !_COMPAT_POLL_H_ */

/*
 * Copyright (c) 2008,2014 Darren Tucker <dtucker@zip.com.au>
 *
 * Permission to use, copy, modify, and distribute this software for any
 * purpose with or without fee is hereby granted, provided that the above
 * copyright notice and this permission notice appear in all copies.
 *
 * THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
 * WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
 * MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
 * ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
 * WHATSOEVER RESULTING FROM LOSS OF MIND, USE, DATA OR PROFITS, WHETHER

```

```

* IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING
* OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*/

#include "includes.h"

#if !defined(HAVE_STATVFS) || !defined(HAVE_FSTATVFS)

#include <sys/types.h>

#ifdef HAVE_SYS_MOUNT_H
#include <sys/mount.h>
#endif
#ifdef HAVE_SYS_STATFS_H
#include <sys/statfs.h>
#endif
#ifdef HAVE_SYS_VFS_H
#include <sys/vfs.h>
#endif

#ifdef HAVE_FSBLKCNT_T
typedef unsigned long fsblkcnt_t;
#endif
#ifdef HAVE_FSFILCNT_T
typedef unsigned long fsfilcnt_t;
#endif

#ifdef ST_RDONLY
#define ST_RDONLY 1
#endif
#ifdef ST_NOSUID
#define ST_NOSUID 2
#endif

/* as defined in IEEE Std 1003.1, 2004 Edition */
struct statvfs {
    unsigned long f_bsize; /* File system block size. */
    unsigned long f_frsize; /* Fundamental file system block size. */
    fsblkcnt_t f_blocks; /* Total number of blocks on file system in */
    /* units of f_frsize. */
    fsblkcnt_t f_bfree; /* Total number of free blocks. */
    fsblkcnt_t f_bavail; /* Number of free blocks available to */
    /* non-privileged process. */
    fsfilcnt_t f_files; /* Total number of file serial numbers. */
    fsfilcnt_t f_ffree; /* Total number of free file serial numbers. */
    fsfilcnt_t f_favail; /* Number of file serial numbers available to */
    /* non-privileged process. */

```

```

unsigned long f_fsid; /*
File system ID. */
unsigned long f_flag; /* BBit mask of f_flag values. */
unsigned long f_namemax; /* Maximum filename length. */
};
#endif

#ifndef HAVE_STATVFS
int statvfs(const char *, struct statvfs *);
#endif

#ifndef HAVE_FSTATVFS
int fstatvfs(int, struct statvfs *);
#endif

/*
 * Copyright (c) 1999-2004 Damien Miller <djm@mindrot.org>
 *
 * Permission to use, copy, modify, and distribute this software for any
 * purpose with or without fee is hereby granted, provided that the above
 * copyright notice and this permission notice appear in all copies.
 *
 * THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
 * WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
 * MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
 * ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
 * WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
 * ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
 * OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
 */

#ifndef _BSD_MISC_H
#define _BSD_MISC_H

#include "includes.h"

char *ssh_get_progname(char *);
int seed_from_prngd(unsigned char *, size_t);

#ifndef HAVE_SETSID
#define setsid() setpggrp(0, getpid())
#endif

/* !HAVE_SETSID */

#ifndef HAVE_SETENV
int setenv(const char *, const char *, int);
#endif /* !HAVE_SETENV */

#ifndef HAVE_SETLOGIN

```

```

int setlogin(const char *);
#endif /* !HAVE_SETLOGIN */

#ifndef HAVE_INNETGR
int innetgr(const char *, const char *, const char *, const char *);
#endif /* HAVE_INNETGR */

#if !defined(HAVE_SETEUID) && defined(HAVE_SETREUID)
int seteuid(uid_t);
#endif /* !defined(HAVE_SETEUID) && defined(HAVE_SETREUID) */

#if !defined(HAVE_SETEGID) && defined(HAVE_SETRESGID)
int setegid(uid_t);
#endif /* !defined(HAVE_SETEGID) && defined(HAVE_SETRESGID) */

#if !defined(HAVE_STRERROR) && defined(HAVE_SYS_ERRLIST) && defined(HAVE_SYS_NERR)
const char *strerror(int);
#endif

#if !defined(HAVE_SETLINEBUF)
#define setlinebuf(a) (setvbuf((a), NULL, _IOLBF, 0))
#endif

#ifndef HAVE_UTIMES
#ifndef HAVE_STRUCT_TIMEVAL
struct timeval {
    long tv_sec;
    long tv_usec;
}
#endif /* HAVE_STRUCT_TIMEVAL */

int utimes(const char *, struct timeval *);
#endif /* HAVE_UTIMES */

#ifndef
    AT_FDCWD
#define AT_FDCWD (-2)
#endif

#ifndef HAVE_FCHMODAT
int fchmodat(int, const char *, mode_t, int);
#endif

#ifndef HAVE_FCHOWNAT
int fchownat(int, const char *, uid_t, gid_t, int);
#endif

#ifndef HAVE_TRUNCATE

```

```

int truncate (const char *, off_t);
#endif /* HAVE_TRUNCATE */

#ifndef HAVE_STRUCT_TIMESPEC
struct timespec {
    time_t tv_sec;
    long tv_nsec;
};
#endif /* !HAVE_STRUCT_TIMESPEC */

#if !defined(HAVE_NANOSLEEP) && !defined(HAVE_NSLEEP)
# include <time.h>
int nanosleep(const struct timespec *, struct timespec *);
#endif

#ifndef HAVE_UTIMENSAT
# include <time.h>
/* start with the high bits and work down to minimise risk of overlap */
# ifndef AT_SYMLINK_NOFOLLOW
# define AT_SYMLINK_NOFOLLOW 0x80000000
# endif
int utimensat(int, const char *, const struct timespec[2], int);
#endif /* !HAVE_UTIMENSAT */

#ifndef HAVE_USLEEP
int usleep(unsigned int useconds);
#endif

#ifndef HAVE_TCGETPGRP
pid_t tcgetpgrp(int);
#endif

#ifndef HAVE_TCSENDBREAK
int tcsendbreak(int, int);
#endif

#ifndef
HAVE_UNSETENV
int unsetenv(const char *);
#endif

#ifndef HAVE_ISBLANK
int isblank(int);
#endif

#ifndef HAVE_GETPGID
pid_t getpgid(pid_t);
#endif

```

```

#ifndef HAVE_PSELECT
int pselect(int, fd_set *, fd_set *, fd_set *, const struct timespec *,
    const sigset_t *);
#endif

#ifndef HAVE_ENDGRENT
# define endgrent() do { } while(0)
#endif

#ifndef HAVE_KRB5_GET_ERROR_MESSAGE
# define krb5_get_error_message krb5_get_err_text
#endif

#ifndef HAVE_KRB5_FREE_ERROR_MESSAGE
# define krb5_free_error_message(a,b) do { } while(0)
#endif

#ifndef HAVE_PLEDGE
int pledge(const char *promises, const char *paths[]);
#endif

/* bsd-err.h */
#ifndef HAVE_ERR
void err(int, const char *, ...) __attribute__((format(printf, 2, 3)));
#endif
#ifndef HAVE_ERRX
void errx(int, const char *, ...) __attribute__((format(printf, 2, 3)));
#endif
#ifndef HAVE_WARN
void warn(const char *, ...) __attribute__((format(printf, 1, 2)));
#endif

#ifndef HAVE_LLABS
long long llabs(long long);
#endif

#if defined(HAVE_DECL_BZERO)
&& HAVE_DECL_BZERO == 0
void bzero(void *, size_t);
#endif

#ifndef HAVE_RAISE
int raise(int);
#endif

#ifndef HAVE_GETSID
pid_t getsid(pid_t);

```

```

#endif

#ifndef HAVE_FLOCK
# define LOCK_SH 0x01
# define LOCK_EX 0x02
# define LOCK_NB 0x04
# define LOCK_UN 0x08
int flock(int, int);
#endif

#ifdef FFLUSH_NULL_BUG
# define fflush(x) (_ssh_compat_fflush(x))
#endif

#ifndef HAVE_LOCALTIME_R
struct tm *localtime_r(const time_t *, struct tm *);
#endif

#ifndef HAVE_REALPATH
#define realpath(x, y) (sftp_realpath((x), (y)))
#endif

#endif /* _BSD_MISC_H */
/*
 * Copyright (c) 2000, 2001, 2011, 2013 Corinna Vinschen <vinschen@redhat.com>
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
 * GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 *
 * Created: Sat Sep 02 12:17:00 2000 cv

```

```

*
* This file contains functions for forcing opened file descriptors to
* binary mode on Windows systems.
*/

#ifndef _BSD_CYGWIN_UTIL_H
#define _BSD_CYGWIN_UTIL_H

#ifdef HAVE_CYGWIN

#undef ERROR

/* Avoid including windows headers. */
typedef void *HANDLE;
#define INVALID_HANDLE_VALUE ((HANDLE) -1)
#define DNLEN 16
#define UNLEN 256

/* Cygwin functions for which declarations are only available when including
   windows headers, so we have to define them here explicitly. */
extern HANDLE cygwin_logon_user (const struct passwd *, const char *);
extern void cygwin_set_impersonation_token (const HANDLE);

#include
<sys/cygwin.h>
#include <io.h>

#define CYGWIN_SSH_PRIVSEP_USER (cygwin_ssh_privsep_user())
const char *cygwin_ssh_privsep_user();

int binary_open(const char *, int , ...);
int check_ntsec(const char *);
char **fetch_windows_environment(void);
void free_windows_environment(char **);
int cygwin_ug_match_pattern_list(const char *, const char *);

#ifndef NO_BINARY_OPEN
#define open binary_open
#endif

#endif /* HAVE_CYGWIN */

#endif /* _BSD_CYGWIN_UTIL_H */
/*
* Copyright (c) 1999-2004 Damien Miller <djm@mindrot.org>
*
* Permission to use, copy, modify, and distribute this software for any
* purpose with or without fee is hereby granted, provided that the above

```



```

* copyright notice and this permission notice appear in all copies.
*
* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
* ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
* ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
* OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*/

```

```

#ifndef _BSD_SIGNAL_H
#define _BSD_SIGNAL_H

```

```

#include "includes.h"

```

```

#include <signal.h>

```

```

#ifndef _NSIG
# ifdef NSIG
#  define _NSIG NSIG
# else
#  define _NSIG 128
# endif
#endif

```

```

#if !defined(HAVE_STRSIGNAL)
char
*strsignal(int);
#endif

```

```

#endif /* _BSD_SIGNAL_H */
/*

```

```

* Copyright (c) 2000,2001 Ben Lindstrom. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
*

```

```

* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,

```

\* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT  
 \* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS  
 OR SERVICES; LOSS OF USE,  
 \* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY  
 \* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT  
 \* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF  
 \* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

\*

\*/

#ifndef \_NEXT\_POSIX\_H

#define \_NEXT\_POSIX\_H

#ifdef HAVE\_NEXT

#include <sys/dir.h>

/\* NGROUPS\_MAX is behind -lposix. Use the BSD version which is NGROUPS \*/

#undef NGROUPS\_MAX

#define NGROUPS\_MAX NGROUPS

/\* NeXT's readdir() is BSD (struct direct) not POSIX (struct dirent) \*/

#define dirent direct

/\* Swap out NeXT's BSD wait() for a more POSIX compliant one \*/

pid\_t posix\_wait(int \*);

#define wait(a) posix\_wait(a)

/\* #ifdef wrapped functions that need defining for clean compiling \*/

pid\_t getppid(void);

void vhangup(void);

int inetgr(const char \*, const char \*, const char \*, const char \*);

/\* TERMCAP \*/

int tcgetattr(int, struct termios \*);

int tcsetattr(int, int,  
const struct termios \*);

int tcsetpgrp(int, pid\_t);

speed\_t cfgetospeed(const struct termios \*);

speed\_t cfgetispeed(const struct termios \*);

int cfsetospeed(struct termios \*, int);

int cfsetispeed(struct termios \*, int);

#endif /\* HAVE\_NEXT \*/

#endif /\* \_NEXT\_POSIX\_H \*/

This file is part of the OpenSSH software.

The licences which components of this software fall under are as follows. First, we will summarize and say that all components are under a BSD licence, or a licence more free than that.

OpenSSH contains no GPL code.

1)

- \* Copyright (c) 1995 Tatu Ylonen <ylo@cs.hut.fi>, Espoo, Finland
- \* All rights reserved
- \*
- \* As far as I am concerned, the code I have written for this software
- \* can be used freely for any purpose. Any derived versions of this
- \* software must be clearly marked as such, and if the derived work is
- \* incompatible with the protocol description in the RFC file, it must be
- \* called by a name other than "ssh" or "Secure Shell".

[Tatu continues]

- \* However, I am not implying to give any licenses to any patents or
  - \* copyrights held by third parties, and the software includes parts that
  - \* are not under my direct control. As far as I know, all included
- \* source code is used in accordance with the relevant license agreements
- \* and can be used freely for any purpose (the GNU license being the most
  - \* restrictive); see below for details.

[However, none of that term is relevant at this point in time. All of these restrictively licenced software components which he talks about have been removed from OpenSSH, i.e.,

- RSA is no longer included, found in the OpenSSL library
- IDEA is no longer included, its use is deprecated
- DES is now external, in the OpenSSL library
- GMP is no longer used, and instead we call BN code from OpenSSL
- Zlib is now external, in a library
- The make-ssh-known-hosts script is no longer included
- TSS has been removed
- MD5 is now external, in the OpenSSL library
- RC4 support has been replaced with ARC4 support from OpenSSL
- Blowfish is now external, in the OpenSSL library

[The licence continues]

Note that any information and cryptographic algorithms used in this software are publicly available on the Internet and at any major bookstore, scientific library, and patent office worldwide. More information can be found e.g. at "<http://www.cs.hut.fi/crypto>".

The legal status of this program is some combination of all these permissions and restrictions. Use only at your own responsibility. You will be responsible for any legal consequences yourself; I am not

making any claims whether possessing or using this is legal or not in your country, and I am not taking any responsibility on your behalf.

## NO WARRANTY

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

3)

ssh-keyscan was contributed by David Mazieres under a BSD-style license.

- \* Copyright 1995, 1996 by David Mazieres <dm@lcs.mit.edu>.

- \*

- \* Modification and redistribution in source and binary forms is
- \* permitted provided that due credit is given to the author and the
- \* OpenBSD project by leaving this copyright notice intact.

4)

The Rijndael implementation by Vincent Rijmen, Antoon Bosselaers and Paulo Barreto is in the public domain and distributed with the following license:

- \* @version 3.0 (December 2000)

- \*

- \* Optimised ANSI C code for the Rijndael cipher (now AES)
- \*
- \* @author Vincent Rijmen <vincent.rijmen@esat.kuleuven.ac.be>

\* @author Antoon Bosselaers <antoon.bosselaers@esat.kuleuven.ac.be>

\* @author Paulo Barreto <paulo.barreto@terra.com.br>

\*

\* This code is hereby placed in the public domain.

\*

\* THIS SOFTWARE IS PROVIDED BY THE AUTHORS "AS IS" AND ANY EXPRESS

\* OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

\* WARRANTIES OF MERCHANTABILITY

AND FITNESS FOR A PARTICULAR PURPOSE

\* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE

\* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR

\* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF

\* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR

\* BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,

\* WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE

\* OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,

\* EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

5)

One component of the ssh source code is under a 3-clause BSD license,  
held by the University of California, since we pulled these parts from  
original Berkeley code.

\* Copyright (c) 1983, 1990, 1992, 1993, 1995

\* The Regents of the University of California. All rights reserved.

\*

\* Redistribution and use in source and binary forms, with

or without

\* modification, are permitted provided that the following conditions

\* are met:

\* 1. Redistributions of source code must retain the above copyright

\* notice, this list of conditions and the following disclaimer.

\* 2. Redistributions in binary form must reproduce the above copyright

\* notice, this list of conditions and the following disclaimer in the

\* documentation and/or other materials provided with the distribution.

\* 3. Neither the name of the University nor the names of its contributors

\* may be used to endorse or promote products derived from this software

\* without specific prior written permission.

\*

\* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND

\* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

\* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

\* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS

BE LIABLE

\* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

\* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS

\* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

\* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT

\* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY  
\* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF  
\* SUCH DAMAGE.

6)

Remaining components of the software are provided under a standard  
2-term BSD licence with the following names as copyright holders:

Markus Friedl  
Theo de Raadt  
Niels Provos  
Dug Song  
Aaron Campbell  
Damien Miller  
Kevin Steves  
Daniel Kouril  
Wesley Griffin  
Per Allansson  
Nils Nordman  
Simon Wilkinson

Portable OpenSSH additionally includes code from the following copyright  
holders, also under the 2-term BSD license:

Ben Lindstrom  
Tim Rice  
Andre  
Lucas  
Chris Adams  
Corinna Vinschen  
Cray Inc.  
Denis Parker  
Gert Doering  
Jakob Schlyter  
Jason Downs  
Juha Yrjölä  
Michael Stone  
Networks Associates Technology, Inc.  
Solar Designer  
Todd C. Miller  
Wayne Schroeder  
William Jones  
Darren Tucker  
Sun Microsystems  
The SCO Group  
Daniel Walsh  
Red Hat, Inc  
Simon Vallet / Genoscope

- \* Redistribution and use in source and binary forms, with or without
- \* modification, are permitted provided that the following conditions
- \* are met:
- \* 1. Redistributions of source code must retain the above copyright
- \* notice, this list of conditions and the following disclaimer.
- \* 2. Redistributions in binary form must reproduce the above copyright
- \* notice, this list of conditions and the following disclaimer in the
- \* documentation and/or other materials provided with the distribution.
- \*
- \* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
- \* IMPLIED WARRANTIES, INCLUDING, BUT

NOT LIMITED TO, THE IMPLIED WARRANTIES

- \* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
- \* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
- \* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
- \* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
- \* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
- \* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
- \* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
- \* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

8) Portable OpenSSH contains the following additional licenses:

a) snprintf replacement

- \* Copyright Patrick Powell 1995
- \* This code is based on code written by Patrick Powell
- \* (papowell@astart.com) It may be used for any purpose as long as this
- \* notice remains intact on all source code distributions

b) Compatibility code  
(openbsd-compat)

Apart from the previously mentioned licenses, various pieces of code  
in the openbsd-compat/ subdirectory are licensed as follows:

Some code is licensed under a 3-term BSD license, to the following  
copyright holders:

Todd C. Miller  
 Theo de Raadt  
 Damien Miller  
 Eric P. Allman  
 The Regents of the University of California  
 Constantin S. Svintsoff  
 Kungliga Tekniska Hgskolan

\* Redistribution and use in source and binary forms, with or without  
\* modification, are permitted provided that the following conditions  
\* are met:

- \* 1. Redistributions of source code must retain the above copyright  
\* notice, this list of conditions and the following disclaimer.
- \* 2. Redistributions in binary form must reproduce the above copyright  
\* notice, this list of conditions and the following disclaimer in the  
\* documentation and/or other materials provided with the distribution.
- \* 3. Neither the name of the University nor the names of its contributors  
\*

may be used to endorse or promote products derived from this software  
\* without specific prior written permission.

\*

\* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND  
\* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
\* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
\* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE  
\* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL  
\* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS  
\* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
\* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT  
\* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY  
\* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF  
\* SUCH DAMAGE.

Some code is licensed under an ISC-style license, to the following  
copyright  
holders:

Internet Software Consortium.

Todd C. Miller

Reyk Floeter

Chad Mynhier

\* Permission to use, copy, modify, and distribute this software for any  
\* purpose with or without fee is hereby granted, provided that the above  
\* copyright notice and this permission notice appear in all copies.  
\*

\* THE SOFTWARE IS PROVIDED "AS IS" AND TODD C. MILLER DISCLAIMS ALL  
\* WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES  
\* OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL TODD C. MILLER BE LIABLE  
\* FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES  
\* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION  
\* OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN  
\* CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Some code is licensed under a MIT-style license to the following  
copyright holders:



Free Software Foundation, Inc.

\* Permission is hereby granted, free of charge,  
to any person obtaining a \*  
\* copy of this software and associated documentation files (the \*  
\* "Software"), to deal in the Software without restriction, including \*  
\* without limitation the rights to use, copy, modify, merge, publish, \*  
\* distribute, distribute with modifications, sublicense, and/or sell \*  
\* copies of the Software, and to permit persons to whom the Software is \*  
\* furnished to do so, subject to the following conditions: \*

\*  
\*  
\* The above copyright notice and this permission notice shall be included \*  
\* in all copies or substantial portions of the Software. \*

\*  
\*  
\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS \*  
\* OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF \*  
\* MERCHANTABILITY, FITNESS FOR  
A PARTICULAR PURPOSE AND NONINFRINGEMENT. \*

\* IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, \*  
\* DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR \*  
\* OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR \*  
\* THE USE OR OTHER DEALINGS IN THE SOFTWARE. \*

\*  
\*  
\* Except as contained in this notice, the name(s) of the above copyright \*  
\* holders shall not be used in advertising or otherwise to promote the \*  
\* sale, use or other dealings in this Software without prior written \*  
\* authorization. \*

\*\*\*\*\*/

The Blowfish cipher implementation is licensed by Niels Provos under  
a 3-clause BSD license:

\* Blowfish - a fast block cipher designed by Bruce Schneier  
\*  
\* Copyright 1997 Niels Provos <provos@physnet.uni-hamburg.de>  
\* All rights reserved.  
\*  
\* Redistribution and use in source and binary forms, with or without  
\* modification, are permitted provided that the following conditions  
\* are met:  
\* 1. Redistributions of source code must retain the above copyright  
\* notice, this list of conditions and the following disclaimer.  
\* 2. Redistributions in binary form must reproduce the above copyright  
\* notice, this list of conditions and the following disclaimer in the  
\* documentation and/or other materials provided with the distribution.  
\* 3. The name of the author may not be used to endorse or promote products  
\* derived from this software without specific prior written permission.

\*  
 \* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR  
 \* IMPLIED WARRANTIES, INCLUDING,  
 BUT NOT LIMITED TO, THE IMPLIED WARRANTIES  
 \* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.  
 \* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,  
 \* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT  
 \* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,  
 \* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY  
 \* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT  
 \* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF  
 \* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Some replacement code is licensed by the NetBSD foundation under a  
 2-clause BSD license:

\* Copyright (c) 2001 The NetBSD Foundation, Inc.  
 \* All rights reserved.  
 \*  
 \* This code is derived from software contributed to The NetBSD Foundation  
 \* by  
 Todd Vierling.

\*  
 \* Redistribution and use in source and binary forms, with or without  
 \* modification, are permitted provided that the following conditions  
 \* are met:  
 \* 1. Redistributions of source code must retain the above copyright  
 \* notice, this list of conditions and the following disclaimer.  
 \* 2. Redistributions in binary form must reproduce the above copyright  
 \* notice, this list of conditions and the following disclaimer in the  
 \* documentation and/or other materials provided with the distribution.  
 \*  
 \* THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS  
 \* ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED  
 \* TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR  
 \* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS  
 \* BE LIABLE FOR ANY DIRECT, INDIRECT,  
 INCIDENTAL, SPECIAL, EXEMPLARY, OR  
 \* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF  
 \* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS  
 \* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN  
 \* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)  
 \* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE  
 \* POSSIBILITY OF SUCH DAMAGE.

The replacement base64 implementation has the following MIT-style  
 licenses:

\* Copyright (c) 1996 by Internet Software Consortium.

\*

\* Permission to use, copy, modify, and distribute this software for any  
\* purpose with or without fee is hereby granted, provided that the above  
\* copyright notice and this permission notice appear in all copies.

\*

\* THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS

\* ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED  
WARRANTIES

\* OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE  
\* CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL  
\* DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR  
\* PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS  
\* ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS  
\* SOFTWARE.

\* Portions Copyright (c) 1995 by International Business Machines, Inc.

\*

\* International Business Machines, Inc. (hereinafter called IBM) grants  
\* permission under its copyrights to use, copy, modify, and distribute this  
\* Software with or without fee, provided that the above copyright notice and  
\* all paragraphs of this notice appear in all copies, and that the name of IBM  
\* not be used in connection with the marketing

of any product incorporating

\* the Software or modifications thereof, without specific, written prior  
\* permission.

\*

\* To the extent it has a right to do so, IBM grants an immunity from suit  
\* under its patents, if any, for the use, sale or manufacture of products to  
\* the extent that such products are used for performing Domain Name System  
\* dynamic updates in TCP/IP networks by means of the Software. No immunity is  
\* granted for any product per se or for any other function of any product.

\*

\* THE SOFTWARE IS PROVIDED "AS IS", AND IBM DISCLAIMS ALL WARRANTIES,  
\* INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A  
\* PARTICULAR PURPOSE. IN NO EVENT SHALL IBM BE LIABLE FOR ANY SPECIAL,  
\* DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER  
ARISING

\* OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE, EVEN  
\* IF IBM IS APPRISED OF THE POSSIBILITY OF SUCH DAMAGES.

-----

\$OpenBSD: LICENCE,v 1.20 2017/04/30 23:26:16 djm Exp \$

/\*

\* Copyright (c) 2012 Darren Tucker (dtucker at zip com au).

\*

\* Permission to use, copy, modify, and distribute this software for any

\* purpose with or without fee is hereby granted, provided that the above  
 \* copyright notice and this permission notice appear in all copies.  
 \*  
 \* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES  
 \* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF  
 \* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR  
 \* ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES  
 \* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN  
 \* ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF  
 \* OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.  
 \*/

```
#ifndef HAVE_SETRESGID
int setresgid(gid_t, gid_t, gid_t);
#endif
#ifndef HAVE_SETRESUID
int setresuid(uid_t, uid_t, uid_t);
#endif
```

## 1.153 scapy 2.6.1-r0

### 1.153.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies  
 of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it

if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in

themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component

itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.



If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals

of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

## NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE,  
THERE IS NO WARRANTY

FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR

THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify  
it under the terms of the GNU

General Public License as published by

the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along  
with this program; if not, write to the Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this  
when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute
it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate  
parts of the General Public License. Of course, the commands you use may  
be called something other than `show w' and `show c'; they could even be  
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your  
school, if any, to sign a "copyright disclaimer" for the program, if  
necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into  
proprietary programs. If your program is a subroutine library, you may  
consider it more useful to permit linking proprietary applications with the  
library. If this  
is what you want to do, use the GNU Lesser General  
Public License instead of this License.

```
{
"testfiles": [
  "test/*.uts",
  "test/scapy/layers/*.uts",
  "test/contrib/automotive/*.uts",
  "test/contrib/automotive/obd/*.uts",
```

```

    "test/contrib/automotive/scanner/*.uts",
    "test/contrib/automotive/gm/*.uts",
    "test/contrib/automotive/bmw/*.uts",
    "test/contrib/automotive/xcp/*.uts",
    "test/contrib/automotive/autosar/*.uts",
    "test/contrib/*.uts"
  ],
  "remove_testfiles": [
    "test/linux.uts",
    "test/windows.uts",
    "test/contrib/automotive/ecu_am.uts",
    "test/contrib/automotive/gm/gmlanutils.uts",
    "test/contrib/isotp_packet.uts",
    "test/contrib/isotpscan.uts",
    "test/contrib/isotp_soft_socket.uts"
  ],
  "onlyfailed": true,
  "preexec": {
    "test/contrib/*.uts": "load_contrib(\"%name%\")",
    "test/cert.uts": "load_layer(\"tls\")",
    "test/ssl2.uts": "load_layer(\"tls\")",
    "test/tls*.uts": "load_layer(\"tls\")"
  },
  "kw_ko": [
    "linux",
    "windows",
    "ipv6",
    "vcan_socket",
    "tun",
    "tap"
  ]
}

```

Creative Commons Attribution-NonCommercial-ShareAlike 2.5

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

## License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE

BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

#### 1. Definitions

- a. "Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.
- b. "Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.
- c. "Licensor" means the individual or entity that offers the Work under the terms of this License.
- d. "Original Author" means the individual or entity who created the Work.
- e. "Work" means the copyrightable work of authorship offered under the terms of this License.
- f. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.
- g. "License Elements" means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, Noncommercial, ShareAlike.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

- a. to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
- b. to create and reproduce Derivative Works;
- c. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;
- d. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works;

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved, including but not limited to the rights set forth in Sections 4(e) and 4(f).

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

- a. You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally

perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder.

You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(d), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(d), as requested.

b. You may distribute, publicly display, publicly perform, or publicly digitally perform a Derivative Work only under the terms of this License, a later version of this License with the same License Elements as this License, or a Creative Commons iCommons license that contains the same License Elements as this License (e.g. Attribution-NonCommercial-ShareAlike 2.5 Japan). You must include a copy of, or the Uniform Resource Identifier for, this License or other license specified in the previous sentence with every copy or phonorecord of each Derivative Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Derivative Works that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder, and You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Derivative Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Derivative Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Derivative Work itself to be made subject to the terms of this License.

c. You may not exercise any of the rights granted to You in Section 3 above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation. The exchange of the Work for other copyrighted works by means of digital file-sharing or otherwise shall not be considered to be intended for or directed toward commercial advantage or private monetary compensation, provided there is no payment of any monetary compensation in connection with the exchange of copyrighted works.

d. If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works,

You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

e. For the avoidance of doubt, where the Work is a musical composition:

i. Performance Royalties Under Blanket Licenses. Licensor reserves the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work if that performance is primarily intended for or directed toward commercial advantage or private monetary compensation.

ii. Mechanical Rights and Statutory Royalties. Licensor reserves the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create

from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions), if Your distribution of such cover version is primarily intended for or directed toward commercial advantage or private monetary compensation.

f. Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor reserves the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions), if Your public digital performance is primarily intended for or directed toward commercial advantage or private monetary compensation.

#### 5. Representations, Warranties and Disclaimer

##### UNLESS OTHERWISE MUTUALLY AGREED

TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### 7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

#### 8. Miscellaneous

a. Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver

or consent shall be in writing and signed by the party to be charged with such waiver or consent.

e.

This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, neither party will use the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time.

Creative Commons may be contacted at <http://creativecommons.org/>.

## 1.154 file 5.46-r2

### 1.154.1 Available under license :

\$File: COPYING,v 1.2 2018/09/09 20:33:28 christos Exp \$  
Copyright (c) Ian F. Darwin 1986, 1987, 1989, 1990, 1991, 1992, 1994, 1995.  
Software written by Ian F. Darwin and others;  
maintained 1994- Christos Zoulas.

This software is not subject to any export provision of the United States Department of Commerce, and may be exported to any country or planet.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice immediately at the beginning of the file, without modification, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR



ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.155 libverto 0.3.2-r2

### 1.155.1 Available under license :

Copyright 2011 Red Hat, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.156 nftables-vim 0\_git20200629-r1

### 1.156.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015 nfnty

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.157 alpine-baselayout 3.7.0-r0

### 1.157.1 Available under license :

```
# Automatically generated by apkbuild-cpan, template 1
# Contributor: Francesco Colista <francesco.colista@gmail.com>
# Maintainer: Francesco Colista <francesco.colista@gmail.com>
pkgname=perl-software-license
_pkgreal=Software-License
pkgver=0.103004
pkgrel=0
pkgdesc="packages that provide templated software licenses"
url="http://search.cpan.org/dist/Software-License/"
arch="noarch"
license="GPL PerlArtistic"
cpandepends="perl-text-template perl-data-section perl-sub-install"
cpanmakedepends=""
depends="$cpandepends"
makedepends="perl-dev $cpanmakedepends"
subpackages="$pkgname-doc"
source="http://search.cpan.org/CPAN/authors/id/R/RJ/RJBS/_pkgreal-$pkgver.tar.gz"

_builddir="$srcdir/_pkgreal-$pkgver"

prepare() {
    cd "$_builddir"
    export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}'`
    PERL_MM_USE_DEFAULT=1 perl Makefile.PL INSTALLDIRS=vendor
}

build() {
    cd "$_builddir"
    export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}'`
    make && make test
```

```

}

package() {
    cd "$_builddir"
    make
    DESTDIR="$pkgdir" install || return 1
    find "$pkgdir" \( -name perllocal.pod -o -name .packlist \) -delete
}

md5sums="0c5fd53d93818566234578b27d667b35 Software-License-0.103004.tar.gz"
# Maintainer: Natanael Copa <ncopa@alpinelinux.org>
pkgname=libart-lgpl
pkgver=2.3.21
pkgrel=5
pkgdesc="A library for high-performance 2D graphics"
url="http://www.levien.com/libart/"
arch="all"
license="GPL"
depends=
makedepends=
subpackages="$pkgname-dev"
source="https://download.gnome.org/sources/libart-lgpl/2.3/libart-lgpl-$pkgver.tar.bz2"
"

_buildddir="$srcdir"/libart-lgpl-$pkgver

prepare() {
    cd "$_builddir"
    update_config_sub || return 1
}

build() {
    cd "$_builddir"
    ./configure \
        --build=$CBUILD \
        --host=$CHOST \
        --prefix=/usr \
        || return 1
    make || return 1
}

package() {
    cd "$_builddir"
    make DESTDIR="$pkgdir" install || return 1
}

sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020 libart-lgpl-2.3.21.tar.bz2"

```

## IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.

Do not use or load this software and any associated materials (collectively, the "Software") until you have carefully read the following terms and conditions. By loading or using the Software, you agree to the terms of this Agreement. If you do not wish to so agree, do not install or use the Software.

### 1. COPYRIGHT NOTICE

Some or all of this work - Copyright 1999-2011, Intel Corp. All rights reserved.

### 2. LICENSE

2.1. This is your license from Intel Corp. under its intellectual property rights. You may have additional license terms from the party that provided you this software, covering your right to use that party's intellectual property rights.

2.2. Intel grants, free of charge, to any person ("Licensee") obtaining a copy of the source code appearing in this file ("Covered Code") an irrevocable, perpetual, worldwide license under Intel's copyrights in the base code distributed originally by Intel ("Original Intel Code") to copy, make derivatives, distribute, use and display any portion of the Covered Code in any form, with the right to sublicense such rights; and

2.3. Intel grants Licensee a non-exclusive and non-transferable patent license (with the right to sublicense), under only those claims of Intel patents that are infringed by the Original Intel Code, to make, use, sell, offer to sell, and import the Covered Code and derivative works thereof solely to the minimum extent necessary to exercise the above copyright license, and in no event shall the patent license extend to any additions to or modifications of the Original Intel Code. No other license or right is granted directly or by implication, estoppel or otherwise; The above copyright and patent license is granted only if the following conditions are met:

### 3. CONDITIONS

3.1. Redistribution of Source with Rights to Further Distribute Source. Redistribution of source code of any substantial portion of the Covered Code or modification with rights to further distribute source must include the above Copyright Notice, the above License, this list of Conditions, and the following Disclaimer and Export Compliance provision. In addition, Licensee must cause all Covered Code to which Licensee contributes to contain a file documenting the

changes Licensee made to create that Covered Code and the date of any change. Licensee must include in that file the documentation of any changes made by any predecessor Licensee. Licensee must include a prominent statement that the modification is derived, directly or indirectly, from Original Intel Code.

3.2. Redistribution of Source with no Rights to Further Distribute Source. Redistribution of source code of any substantial portion of the Covered Code or modification without rights to further distribute source must include the following Disclaimer and Export Compliance provision in the documentation and/or other materials provided with distribution. In addition,

Licensee may not authorize further sublicense of source of any portion of the Covered Code, and must include terms to the effect that the license from Licensee to its licensee is limited to the intellectual property embodied in the software Licensee provides to its licensee, and not to intellectual property embodied in modifications its licensee may make.

3.3. Redistribution of Executable. Redistribution in executable form of any substantial portion of the Covered Code or modification must reproduce the above Copyright Notice, and the following Disclaimer and Export Compliance provision in the documentation and/or other materials provided with the distribution.

3.4. Intel retains all right, title, and interest in and to the Original Intel Code.

3.5. Neither the name Intel nor any other trademark owned or controlled by Intel shall be used in advertising or otherwise to promote the sale, use or other dealings in products derived from or relating to the Covered Code without prior written authorization from Intel.

#### 4. DISCLAIMER AND EXPORT COMPLIANCE

4.1. INTEL MAKES NO WARRANTY OF ANY KIND REGARDING ANY SOFTWARE PROVIDED HERE. ANY SOFTWARE ORIGINATING FROM INTEL OR DERIVED FROM INTEL SOFTWARE IS PROVIDED "AS IS," AND INTEL WILL NOT PROVIDE ANY SUPPORT, ASSISTANCE, INSTALLATION, TRAINING OR OTHER SERVICES. INTEL WILL NOT PROVIDE ANY UPDATES, ENHANCEMENTS OR EXTENSIONS. INTEL SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE.

4.2. IN NO EVENT SHALL INTEL HAVE ANY LIABILITY TO LICENSEE, ITS LICENSEES OR ANY OTHER THIRD PARTY, FOR ANY LOST PROFITS, LOST DATA, LOSS OF USE OR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES,

OR FOR ANY INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, UNDER ANY CAUSE OF ACTION OR THEORY OF LIABILITY, AND IRRESPECTIVE OF WHETHER INTEL HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES. THESE LIMITATIONS SHALL APPLY NOTWITHSTANDING THE FAILURE OF THE ESSENTIAL PURPOSE OF ANY LIMITED REMEDY.

4.3. Licensee shall not export, either directly or indirectly, any of this software or system incorporating such software without first obtaining any required license or other approval from the U. S. Department of Commerce or any other agency or department of the United States Government. In the event Licensee exports any such software from the United States or re-exports any such software from a foreign destination, Licensee shall ensure that the distribution and export/re-export of the software is in compliance with all laws, regulations, orders, or other restrictions of the U.S. Export Administration Regulations. Licensee agrees that neither it nor any of its subsidiaries will export/re-export any technical data, process, software, or service, directly or indirectly, to any country for which the United States government or any agency thereof requires an export license, other governmental approval, or letter of assurance, without first obtaining such license, approval or letter.

#### INTEL SOFTWARE LICENSE AGREEMENT

#### IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.

Do not use or load this software and any associated materials (collectively, the "Software") until you have carefully read the following terms and conditions. By loading or using the Software, you agree to the terms of this Agreement. If you do not wish to so agree, do not install or use the Software.

#### LICENSES: Please Note:

- If you are a network administrator, the "Site License" below shall apply to you.
- If you are an end user, the "Single User License" shall apply to you.
- If you are an original equipment manufacturer (OEM), the "OEM License" shall apply to you.

**SITE LICENSE.** You may copy the Software onto your organization's computers for your organization's use, and you may make a reasonable number of back-up copies of the Software, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part of the Software except as provided in this Agreement, and you agree to

prevent unauthorized copying of the Software.

3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may not sublicense or permit simultaneous use of the Software by more than one user.
5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

**SINGLE USER LICENSE.** You may copy the Software onto a single computer for your personal, noncommercial use, and you may make one back-up copy of the Software, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part of the Software except as provided in this Agreement, and you agree to prevent unauthorized copying of the Software.
3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may not sublicense or permit simultaneous use of the Software by more than one user.
5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

**OEM LICENSE:** You may reproduce and distribute the Software only as an integral part of or incorporated in Your product or as a standalone Software maintenance update for existing end users of Your products, excluding any other standalone products, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part of the Software except as provided in this Agreement, and you agree to prevent unauthorized copying of the Software.
3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may only distribute the Software to your customers pursuant to a written license agreement. Such license agreement may be a "break-the-seal" license agreement. At a minimum such license shall safeguard Intel's ownership rights to the Software.
5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

**NO OTHER RIGHTS.** No rights or licenses are granted by Intel to You, expressly or by implication, with respect to any proprietary information or patent, copyright, mask work, trademark, trade secret, or other intellectual property right owned or controlled by Intel, except as expressly provided in this

Agreement.

OWNERSHIP OF SOFTWARE AND COPYRIGHTS. Title to all copies of the Software remains with Intel or its suppliers. The Software is copyrighted and protected

by the laws of the United States and other countries, and international treaty provisions. You may not remove any copyright notices from the Software. Intel may make changes to the Software, or to items referenced therein, at any time without notice, but is not obligated to support or update the Software. Except as otherwise expressly provided, Intel grants no express or implied right under Intel patents, copyrights, trademarks, or other intellectual property rights. You may transfer the Software only if the recipient agrees to be fully bound by these terms and if you retain no copies of the Software.

LIMITED MEDIA WARRANTY. If the Software has been delivered by Intel on physical media, Intel warrants the media to be free from material physical defects for a period of ninety days after delivery by Intel. If such a defect is found, return the media to Intel for replacement or alternate delivery of the Software as Intel may select.

EXCLUSION OF OTHER WARRANTIES. EXCEPT AS PROVIDED ABOVE, THE SOFTWARE IS PROVIDED "AS IS" WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. Intel does not warrant or assume responsibility for the accuracy or completeness of any information, text, graphics, links or other items contained within the Software.

LIMITATION OF LIABILITY. IN NO EVENT SHALL INTEL OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, OR LOST INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF INTEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME JURISDICTIONS PROHIBIT EXCLUSION OR LIMITATION OF LIABILITY FOR IMPLIED WARRANTIES OR CONSEQUENTIAL OR INCIDENTAL DAMAGES, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU. YOU MAY ALSO HAVE OTHER LEGAL RIGHTS THAT VARY FROM JURISDICTION TO JURISDICTION.

TERMINATION OF THIS AGREEMENT. Intel may terminate this Agreement at any time if you violate its terms. Upon termination, you will immediately destroy the Software or return all copies of the Software to Intel.

APPLICABLE LAWS. Claims arising under this Agreement shall be governed by the laws of California, excluding its principles of conflict of laws and the United Nations Convention on Contracts for the Sale of Goods. You may not export the Software in violation of applicable export laws and regulations. Intel is not obligated under any other agreements unless they are in writing



and signed by an authorized representative of Intel.

GOVERNMENT RESTRICTED RIGHTS. The Software is provided with "RESTRICTED RIGHTS." Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor. Use of the Software by the Government constitutes acknowledgment of Intel's proprietary rights therein. Contractor or Manufacturer is Intel  
2200 Mission College Blvd., Santa Clara, CA  
95052.

/\*

\* Copyright (c) 1983, 1988 Regents of the University of California.

\* All rights reserved.

\*

\* Redistribution and use in source and binary forms are permitted

\* provided that the above copyright notice and this paragraph are

\* duplicated in all such forms and that any documentation,

\* advertising materials, and other materials related to such

\* distribution and use acknowledge that the software was developed

\* by the University of California, Berkeley. The name of the

\* University may not be used to endorse or promote products derived

\* from this software without specific prior written permission.

\* THIS SOFTWARE IS PROVIDED ``AS IS" AND WITHOUT ANY EXPRESS OR

\* IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED

\* WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

\*/

# Automatically generated by apkgbuild-cpan, template 1

# Contributor: Valery Kartel <valery.kartel@gmail.com>

# Maintainer: Valery Kartel <valery.kartel@gmail.com>

pkgname=perl-bsd-resource

\_pkgreal=BSD-Resource

pkgver=1.2909

pkgrel=2

pkgdesc="Perl extension implements the BSD process resource limit functions"

url="http://search.cpan.org/dist/BSD-Resource/"

arch="all"

license="GPL PerlArtistic"

cpandepends=""

cpanmakedepends=""

depends="\$cpandepends"

makedepends="perl-dev \$cpanmakedepends"

subpackages="\$pkgname-doc"

source="http://search.cpan.org/CPAN/authors/id/J/JH/JHI/\_pkgreal-\$pkgver.tar.gz"

\_builddir="\$srcdir/\_pkgreal-\$pkgver"

prepare() {

cd "\$\_builddir"

export CFLAGS=`perl -MConfig -E 'say \$Config{ccflags}`

```

PERL_MM_USE_DEFAULT=1 perl Makefile.PL INSTALLDIRS=vendor
}

build() {
    cd "$_builddir"
    export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}'`
    make && make test || return 1
}

package() {
    cd "$_builddir"
    make DESTDIR="$pkgdir" install || return 1
    find "$pkgdir"
    \ ( -name perllocal.pod -o -name .packlist ) -delete
}

md5sums="3c6dd2fa953088aa87e263f40818d012 BSD-Resource-1.2909.tar.gz"
sha256sums="c47d460e2bd00d8d049f6ff4a0f1f61957e289e8de45af5fc79e851e2e855e06 BSD-Resource-1.2909.tar.gz"
sha512sums="5f489eaac5089e82081b7cf51d62638750e443751b49415407354da96b17771819b5e6e85529faa41ddd85743363750157943c7f7700276cce2bd40a429fa430 BSD-Resource-1.2909.tar.gz"
SQLite Copyright
SQLite is in the
Public Domain

```

All of the deliverable code in SQLite has been dedicated to the public domain by the authors. All code authors, and representatives of the companies they work for, have signed affidavits dedicating their contributions to the public domain and originals of those signed affidavits are stored in a firesafe at the main offices of Hwaci. Anyone is free to copy, modify, publish, use, compile, sell, or distribute the original SQLite code, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

The previous paragraph applies to the deliverable code in SQLite - those parts of the SQLite library that you actually bundle and ship with a larger application. Portions of the documentation and some code used as part of the build process might fall under other licenses. The details here are unclear. We do not worry about the licensing of the documentation and build code so much because none of these things are part of the core deliverable SQLite library.

All of the deliverable code in SQLite has been written from scratch. No code has been taken from other projects or from the open internet. Every line of code can be traced back to its original author, and all of those authors have public domain dedications on file. So the SQLite code base is clean and is uncontaminated with licensed code from other projects.

#### Obtaining An Explicit License To Use SQLite

Even though SQLite is in the public domain and does not require a license, some users want to obtain a license anyway. Some reasons for obtaining a license include:

You are using SQLite in a jurisdiction that does not recognize the public domain.

You are using SQLite in a jurisdiction that does not recognize the right of an author to dedicate their work to the

public domain.

You want to hold a tangible legal document as evidence that you have the legal right to use and distribute SQLite.  
Your legal department tells  
you that you have to purchase a license.

If you feel like you really have to purchase a license for SQLite, Hwaci, the company that employs the architect and principal developers of SQLite, will sell you one.

#### Contributed Code

In order to keep SQLite completely free and unencumbered by copyright, all new contributors to the SQLite code base are asked to dedicate their contributions to the public domain. If you want to send a patch or enhancement for possible inclusion in the SQLite source tree, please accompany the patch with the following statement:

The author or authors of this code dedicate any and all copyright interest in this code to the public domain. We make this dedication for the benefit of the public at large and to the detriment of our heirs and successors. We intend this dedication to be an overt act of relinquishment in perpetuity of all present and future rights to this code under copyright law.

We are not able to accept patches or changes to SQLite that are not accompanied by a statement such as the above. In addition, if you make changes or enhancements as an employee, then a simple statement such as the above is insufficient. You must also send by surface mail a copyright release signed by a company officer. A signed original of the copyright release should be mailed to:

Hwaci  
6200 Maple Cove Lane  
Charlotte, NC 28269  
USA

A template copyright release is available in PDF or HTML. You can use this release to make future changes.

see <http://www.sqlite.org/copyright.html>

Copyright (c) 2016 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software source and associated documentation files (the "Materials"), to deal in the Materials without restriction, including without limitation the rights to use, copy, modify, compile, merge, publish, distribute, sublicense, and/or sell copies of the Materials, and to permit persons to whom the Materials are furnished to do so, subject the following terms and conditions:

All modifications to the Materials used to create a binary that is distributed to third parties shall be provided to Khronos with an unrestricted license to use for the purposes of implementing bug fixes and enhancements to the Materials;

If the binary is used as part of an OpenCL(TM) implementation, whether binary is distributed together with or separately to that implementation, then recipient must become an OpenCL Adopter and follow the published OpenCL conformance process for that implementation,

details at:  
<http://www.khronos.org/conformance/>;

The above copyright notice, the OpenCL trademark license, and this permission notice shall be included in all copies or substantial portions of the Materials.

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.

OpenCL is a trademark of Apple Inc. used under license by Khronos.

# Contributor: Fabian Affolter <fabian@affolter-engineering.ch>

# Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>

pkgname=py-flake8-copyright

\_pkgname=flake8-copyright

pkgver=0.2.0

pkgrel=1

pkgdesc="Extension for flake8 which checks for copyrights"

url="https://github.com/savoirfairelinux/flake8-copyright"

arch="noarch"

license="MIT"

depends="flake8"

makedepends="python3-dev"

source="https://files.pythonhosted.org/packages/source/\${\_pkgname:0:1}/\${\_pkgname}/\${\_pkgname}-\${pkgver}.tar.gz"

builddir="\$srcdir"/\$\_pkgname-\$pkgver

```
build() {  
    cd "$builddir"  
    python3 setup.py build || return 1  
}
```

```
package() {  
    cd "$builddir"  
    python3 setup.py install --prefix=/usr --root="$pkgdir" || return 1  
}
```

md5sums="c6348990c6d74933bf5d5f6a55a14bdf flake8-copyright-0.2.0.tar.gz"

sha256sums="aeef26eb4d5223c9cd5b101e68175fcef6d2b353bf36da688fdde62fccfe2b73 flake8-copyright-0.2.0.tar.gz"

sha512sums="bab1bfaeed3a525ac3c7da12b1f2ec338c204cc69b72d05cd39560e528faff5d586c3b1be9570f4a977a7e2417b586e543bfba12abce256b2023c8c4993eeb90

flake8-copyright-0.2.0.tar.gz"

mini\_sendmail - accept email on behalf of real sendmail

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.158 conntrack-tools 1.4.8-r0

### 1.158.1 Available under license :

GNU GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
675 Mass Ave, Cambridge, MA 02139, USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by

the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work

which contains  
a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim  
copies of the Program's  
source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole

or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide

a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code



or executable form with such  
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy  
the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6.

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent

infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number

of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest

to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

## 1.159 brotli 1.1.0-r2

### 1.159.1 Available under license :

Copyright (c) 2009, 2010, 2013-2016 by the Brotli Authors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.160 llvm 18.1.8-r5

### 1.160.1 Available under license :

; NOTE: Assertions have been autogenerated by utils/update\_llc\_test\_checks.py

; RUN: llc < %s -mtriple=aarch64-- | FileCheck %s

; A shuffle mask with all undef elements is always legal.

```
define <4 x i32> @PR41535(<2 x i32> %p1, <2 x i32> %p2) {
; CHECK-LABEL: PR41535:
; CHECK:      // %bb.0:
; CHECK-NEXT:  ext v0.8b, v0.8b, v1.8b, #4
; CHECK-NEXT:  mov v0.d[1], v0.d[0]
; CHECK-NEXT:  ret
%cat1 = shufflevector <2 x i32> %p1, <2 x i32> undef, <4 x i32> <i32 undef, i32 1, i32 undef, i32 undef>
%cat2 = shufflevector <2 x i32> %p2, <2 x i32> undef, <4 x i32> <i32 0, i32 undef, i32 undef, i32 undef>
```

```

%r = shufflevector <4 x i32> %cat1, <4 x i32> %cat2, <4 x i32> <i32 undef, i32 undef, i32 1, i32 4>
ret <4 x i32> %r
}
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=amdgc -mcpu=fiji < %s | FileCheck -enable-var-scope --check-prefixes=GCN,GCN-
SAFE,VI,VI-SAFE %s
; RUN: llc -enable-no-signed-zeros-fp-math -mtriple=amdgc -mcpu=fiji < %s | FileCheck -enable-var-scope --
check-prefixes=GCN,GCN-NSZ,VI,VI-NSZ %s

; -----
; rcp tests
; -----

define half @v_fneg_rcp_f16(half %a) #0 {
; GCN-LABEL: v_fneg_rcp_f16:
; GCN:      ; %bb.0:
; GCN-NEXT:  s_waitcnt vmcnt(0) expcnt(0) lgkmcnt(0)
; GCN-NEXT:  v_rcp_f16_e64 v0, -v0
; GCN-NEXT:  s_setpc_b64 s[30:31]
%rcp = call half @llvm.amdgc.rcp.f16(half %a)
%fneg = fneg half %rcp
ret half %fneg
}

define half @v_fneg_rcp_fneg_f16(half %a) #0 {
; GCN-LABEL: v_fneg_rcp_fneg_f16:
; GCN:      ; %bb.0:
; GCN-NEXT:  s_waitcnt vmcnt(0) expcnt(0) lgkmcnt(0)
; GCN-NEXT:
    v_rcp_f16_e32 v0, v0
; GCN-NEXT:  s_setpc_b64 s[30:31]
%fneg.a = fneg half %a
%rcp = call half @llvm.amdgc.rcp.f16(half %fneg.a)
%fneg = fneg half %rcp
ret half %fneg
}

define { half, half } @v_fneg_rcp_store_use_fneg_f16(half %a) #0 {
; GCN-LABEL: v_fneg_rcp_store_use_fneg_f16:
; GCN:      ; %bb.0:
; GCN-NEXT:  s_waitcnt vmcnt(0) expcnt(0) lgkmcnt(0)
; GCN-NEXT:  v_rcp_f16_e32 v2, v0
; GCN-NEXT:  v_xor_b32_e32 v1, 0x8000, v0
; GCN-NEXT:  v_mov_b32_e32 v0, v2
; GCN-NEXT:  s_setpc_b64 s[30:31]
%fneg.a = fneg half %a
%rcp = call half @llvm.amdgc.rcp.f16(half %fneg.a)
%fneg = fneg half %rcp

```

```

%insert.0 = insertvalue { half, half } poison, half %fneg, 0
%insert.1 = insertvalue { half, half } %insert.0, half %fneg.a, 1
ret { half, half } %insert.1
}

```

```

define { half, half } @v_fneg_rcp_multi_use_fneg_f16(half %a, half %c) #0 {
; GCN-LABEL: v_fneg_rcp_multi_use_fneg_f16:
; GCN:      ; %bb.0:
; GCN-NEXT:  s_waitcnt vmcnt(0) expcnt(0) lgkmcnt(0)
; GCN-NEXT:
    v_rcp_f16_e32 v2, v0
; GCN-NEXT:  v_mul_f16_e64 v1, -v0, v1
; GCN-NEXT:  v_mov_b32_e32 v0, v2
; GCN-NEXT:  s_setpc_b64 s[30:31]
    %fneg.a = fneg half %a
    %rcp = call half @llvm.amdgcn.rcp.f16(half %fneg.a)
    %fneg = fneg half %rcp
    %use1 = fmul half %fneg.a, %c
    %insert.0 = insertvalue { half, half } poison, half %fneg, 0
    %insert.1 = insertvalue { half, half } %insert.0, half %use1, 1
    ret { half, half } %insert.1
}

```

```

; -----
; sin tests
; -----

```

```

define half @v_fneg_amdgcn_sin_f16(half %a) #0 {
; GCN-LABEL: v_fneg_amdgcn_sin_f16:
; GCN:      ; %bb.0:
; GCN-NEXT:  s_waitcnt vmcnt(0) expcnt(0) lgkmcnt(0)
; GCN-NEXT:  v_sin_f16_e64 v0, -v0
; GCN-NEXT:  s_setpc_b64 s[30:31]
    %sin = call half @llvm.amdgcn.sin.f16(half %a)
    %fneg = fneg half %sin
    ret half %fneg
}

```

```

; -----
;
; vintp tests
; -----

```

```

define { float, float } @v_fneg_interp_p1_f16(float %a, float %b) #0 {
; GCN-LABEL: v_fneg_interp_p1_f16:
; GCN:      ; %bb.0:
; GCN-NEXT:  s_waitcnt vmcnt(0) expcnt(0) lgkmcnt(0)
; GCN-NEXT:  v_mul_f32_e64 v1, v0, -v1

```

```

; GCN-NEXT: s_mov_b32 m0, 0
; GCN-NEXT: s_setreg_imm32_b32 hwreg(HW_REG_MODE, 2, 2), 3
; GCN-NEXT: v_interp_p1ll_f16 v0, v1, attr0.x
; GCN-NEXT: v_interp_p1ll_f16 v1, v1, attr0.y
; GCN-NEXT: s_setpc_b64 s[30:31]
%mul = fmul float %a, %b
%fneg = fneg float %mul
%intrp0 = call float @llvm.amdgcn.interp.p1.f16(float %fneg, i32 0, i32 0, i1 false, i32 0)
%intrp1 = call float @llvm.amdgcn.interp.p1.f16(float %fneg, i32 1, i32 0, i1 false, i32 0)
%insert.0 = insertvalue { float, float } poison, float %intrp0, 0
%insert.1 = insertvalue { float, float } %insert.0, float %intrp1, 1
ret { float, float } %insert.1
}

define
{ half, half } @v_fneg_interp_p2_f16(float %a, float %b) #0 {
; GCN-LABEL: v_fneg_interp_p2_f16:
; GCN:      ; %bb.0:
; GCN-NEXT: s_waitcnt vmcnt(0) expcnt(0) lgkmcnt(0)
; GCN-NEXT: v_mul_f32_e64 v1, v0, -v1
; GCN-NEXT: v_mov_b32_e32 v2, 4.0
; GCN-NEXT: s_mov_b32 m0, 0
; GCN-NEXT: s_setreg_imm32_b32 hwreg(HW_REG_MODE, 2, 2), 3
; GCN-NEXT: v_interp_p2_f16 v0, v1, attr0.x, v2
; GCN-NEXT: v_interp_p2_f16 v1, v1, attr0.y, v2
; GCN-NEXT: s_setpc_b64 s[30:31]
%mul = fmul float %a, %b
%fneg = fneg float %mul
%intrp0 = call half @llvm.amdgcn.interp.p2.f16(float 4.0, float %fneg, i32 0, i32 0, i1 false, i32 0)
%intrp1 = call half @llvm.amdgcn.interp.p2.f16(float 4.0, float %fneg, i32 1, i32 0, i1 false, i32 0)
%insert.0 = insertvalue { half, half } poison, half %intrp0, 0
%insert.1 = insertvalue { half, half } %insert.0, half %intrp1, 1
ret { half, half } %insert.1
}

; -----
;
; arithmetic.fence tests
; -----

; FIXME: Legalization/promote is broken
define half @v_fneg_arithmetic_fence_f16(half %a) #0 {
; GCN-LABEL: v_fneg_arithmetic_fence_f16:
; GCN:      ; %bb.0:
; GCN-NEXT: ; ARITH_FENCE
; GCN-NEXT: s_waitcnt vmcnt(0) expcnt(0) lgkmcnt(0)
; GCN-NEXT: v_xor_b32_e32 v0, 0x8000, v0
; GCN-NEXT: s_setpc_b64 s[30:31]

```



```

%fence = call half @llvm.arithmetic.fence.f16(half %a)
%fneg = fneg half %fence
ret half %fneg
}

define half @v_fneg_arithmetic_fence_fmulo_f16(half %a, half %b) #0 {
; GCN-LABEL: v_fneg_arithmetic_fence_fmulo_f16:
; GCN:      ; %bb.0:
; GCN-NEXT:  s_waitcnt vmcnt(0) expcnt(0) lgkmcnt(0)
; GCN-NEXT:  v_mul_f16_e32 v0, v0, v1
; GCN-NEXT:  ; ARITH_FENCE
; GCN-NEXT:  v_xor_b32_e32 v0, 0x8000, v0
; GCN-NEXT:  s_setpc_b64 s[30:31]
%mul = fmul half %a, %b
%fence = call half @llvm.arithmetic.fence.f16(half %mul)
%fneg = fneg half %fence
ret half
%fneg
}

declare half @llvm.amdgcn.rcp.f16(half) #1
declare half @llvm.amdgcn.sin.f16(half) #1
declare half @llvm.arithmetic.fence.f16(half) #1
declare float @llvm.amdgcn.interp.p1.f16(float, i32, i32, i1, i32) #0
declare half @llvm.amdgcn.interp.p2.f16(float, float, i32, i32, i1, i32) #0

attributes #0 = { nounwind "denormal-fp-math-f32"="preserve-sign,preserve-sign" }
attributes #1 = { nounwind readnone }
attributes #2 = { nounwind "unsafe-fp-math"="true" }
attributes #3 = { nounwind "no-signed-zeros-fp-math"="true" }
attributes #4 = { nounwind "amdgp-ieee"="false" "denormal-fp-math-f32"="preserve-sign,preserve-sign" }
;; NOTE: These prefixes are unused and the list is autogenerated. Do not add tests below this line:
; GCN-NSZ: {{.*}}
; GCN-SAFE: {{.*}}
; VI: {{.*}}
; VI-NSZ: {{.*}}
; VI-SAFE: {{.*}}
; NOTE: Assertions have been autogenerated by utils/update_test_checks.py UTC_ARGS: --function-signature --
check-attributes --check-globals
; RUN: opt -aa-pipeline=basic-aa -passes=attributor -attributor-manifest-internal -attributor-annotate-decl-cs -S <
%s | FileCheck %s --check-prefixes=CHECK,TUNIT
; RUN: opt -aa-pipeline=basic-aa -passes=attributor-cgsc -attributor-manifest-internal -attributor-annotate-decl-cs -
S < %s | FileCheck %s --check-prefixes=CHECK,CGSCC
; Test that we only promote arguments when the caller/callee have compatible
; function attributes.

target triple = "x86_64-unknown-linux-gnu"

```

```

; This should promote
define internal fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512(ptr %arg, ptr
readonly %arg1) #0 {
;
; CHECK: Function Attrs: inlinehint mustprogress norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CHECK-LABEL: define {{^@+}}@callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512
; CHECK-SAME:
(ptr noalias nocapture norecurse noundef nonnull writeonly align 64 dereferenceable(64) [[ARG:%.*]], <8 x i64>
[[TMP0:%.*]]) #[[ATTR0:[0-9]+]] {
; CHECK-NEXT: bb:
; CHECK-NEXT: [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
; CHECK-NEXT: store <8 x i64> [[TMP0]], ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT: [[TMP:%.*]] = load <8 x i64>, ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT: store <8 x i64> [[TMP]], ptr [[ARG]], align 64
; CHECK-NEXT: ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

define void @avx512_legal512_prefer512_call_avx512_legal512_prefer512(ptr %arg) #0 {
;
; TUNIT: Function Attrs: inlinehint mustprogress norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; TUNIT-LABEL: define {{^@+}}@avx512_legal512_prefer512_call_avx512_legal512_prefer512
; TUNIT-SAME: (ptr nocapture norecurse writeonly [[ARG:%.*]]) #[[ATTR0]] {
; TUNIT-NEXT: bb:
; TUNIT-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align
32
; TUNIT-NEXT: [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT: call void @llvm.memset.p0.i64(ptr noalias nocapture norecurse noundef nonnull writeonly align 32
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5:[0-9]+]]
; TUNIT-NEXT: [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64
; TUNIT-NEXT: call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512(ptr
noalias nocapture norecurse noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64>
[[TMP0]]) #[[ATTR6:[0-9]+]]
; TUNIT-NEXT: [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; TUNIT-NEXT: store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; TUNIT-NEXT: ret void
;
; CGSCC: Function Attrs: inlinehint mustprogress norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CGSCC-LABEL: define {{^@+}}@avx512_legal512_prefer512_call_avx512_legal512_prefer512
; CGSCC-SAME: (ptr nocapture

```

```

nofree noundef nonnull writeonly align 2 dereferenceable(64) [[ARG:%.*]]) #[[ATTR0]] {
; CGSCC-NEXT: bb:
; CGSCC-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT: [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT: call void @llvm.memset.p0.i64(ptr noalias nocapture nofree noundef nonnull writeonly align 64
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5:[0-9]+]]
; CGSCC-NEXT: [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64
; CGSCC-NEXT: call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512(ptr
noalias nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64>
[[TMP0]]) #[[ATTR6:[0-9]+]]
; CGSCC-NEXT: [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; CGSCC-NEXT: store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; CGSCC-NEXT: ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align 32
call void
@llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512(ptr %tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

; This should promote
define internal fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256(ptr %arg, ptr
readonly %arg1) #1 {
;
; CHECK: Function Attrs: inlinehint mustprogress nofree norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CHECK-LABEL: define {{^[^@]+}}@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256
; CHECK-SAME: (ptr noalias nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64)
[[ARG:%.*]], <8 x i64> [[TMP0:%.*]]) #[[ATTR1:[0-9]+]] {
; CHECK-NEXT: bb:
; CHECK-NEXT: [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
; CHECK-NEXT: store <8 x i64> [[TMP0]], ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT: [[TMP:%.*]] = load <8 x
i64>, ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT: store <8 x i64> [[TMP]], ptr [[ARG]], align 64
; CHECK-NEXT: ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

```

```

define void @avx512_legal512_prefer256_call_avx512_legal512_prefer256(ptr %arg) #1 {
;
; TUNIT: Function Attrs: inlinehint mustprogress norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; TUNIT-LABEL: define {{^@+}}@avx512_legal512_prefer256_call_avx512_legal512_prefer256
; TUNIT-SAME: (ptr nocapture norecurse writeonly [[ARG:%.*]]) #[[ATTR1]] {
; TUNIT-NEXT: bb:
; TUNIT-NEXT:  [[TMP:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT:  [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT:  call void @llvm.memset.p0.i64(ptr noalias nocapture norecurse noundef nonnull writeonly align 32
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5]]
; TUNIT-NEXT:  [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64
; TUNIT-NEXT:
    call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256(ptr noalias nocapture
norecurse noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64> [[TMP0]]) #[[ATTR6]]
; TUNIT-NEXT:  [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; TUNIT-NEXT:  store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; TUNIT-NEXT:  ret void
;
; CGSCC: Function Attrs: inlinehint mustprogress norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CGSCC-LABEL: define {{^@+}}@avx512_legal512_prefer256_call_avx512_legal512_prefer256
; CGSCC-SAME: (ptr nocapture norecurse noundef nonnull writeonly align 2 dereferenceable(64) [[ARG:%.*]])
#[[ATTR1]] {
; CGSCC-NEXT: bb:
; CGSCC-NEXT:  [[TMP:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT:  [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT:  call void @llvm.memset.p0.i64(ptr noalias nocapture norecurse noundef nonnull writeonly align 64
dereferenceable(64)
[[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5]]
; CGSCC-NEXT:  [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64
; CGSCC-NEXT:  call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256(ptr
noalias nocapture norecurse noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64>
[[TMP0]]) #[[ATTR6]]
; CGSCC-NEXT:  [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; CGSCC-NEXT:  store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; CGSCC-NEXT:  ret void
;
bb:
    %tmp = alloca <8 x i64>, align 32
    %tmp2 = alloca <8 x i64>, align 32
    call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
    call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256(ptr %tmp2, ptr %tmp)
    %tmp4 = load <8 x i64>, ptr %tmp2, align 32
    store <8 x i64> %tmp4, ptr %arg, align 2
    ret void
}

```

```

; This should promote
define internal fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256(ptr
    %arg, ptr readonly %arg1) #1 {
;
; CHECK: Function Attrs: inlinehint mustprogress norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CHECK-LABEL: define {{^[^@]+}}@callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256
; CHECK-SAME: (ptr noalias nocapture norecurse noundef nonnull writeonly align 64 dereferenceable(64)
[[ARG:%.*]], <8 x i64> [[TMP0:%.*]]) #1[[ATTR1]] {
; CHECK-NEXT: bb:
; CHECK-NEXT: [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
; CHECK-NEXT: store <8 x i64> [[TMP0]], ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT: [[TMP:%.*]] = load <8 x i64>, ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT: store <8 x i64> [[TMP]], ptr [[ARG]], align 64
; CHECK-NEXT: ret void
;
bb:
    %tmp = load <8 x i64>, ptr %arg1
    store <8 x i64> %tmp, ptr %arg
    ret void
}

define void @avx512_legal512_prefer512_call_avx512_legal512_prefer256(ptr %arg) #0 {
;
; TUNIT: Function Attrs: inlinehint mustprogress norecurse nosync nounwind
willreturn memory(argmem: readwrite) uwtable
; TUNIT-LABEL: define {{^[^@]+}}@avx512_legal512_prefer512_call_avx512_legal512_prefer256
; TUNIT-SAME: (ptr nocapture norecurse writeonly [[ARG:%.*]]) #1[[ATTR0]] {
; TUNIT-NEXT: bb:
; TUNIT-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT: [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT: call void @llvm.memset.p0.i64(ptr noalias nocapture norecurse noundef nonnull writeonly align 32
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #1[[ATTR5]]
; TUNIT-NEXT: [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64
; TUNIT-NEXT: call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256(ptr
noalias nocapture norecurse noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64>
[[TMP0]]) #1[[ATTR6]]
; TUNIT-NEXT: [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; TUNIT-NEXT: store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; TUNIT-NEXT:
    ret void
;
; CGSCC: Function Attrs: inlinehint mustprogress norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CGSCC-LABEL: define {{^[^@]+}}@avx512_legal512_prefer512_call_avx512_legal512_prefer256
; CGSCC-SAME: (ptr nocapture norecurse noundef nonnull writeonly align 2 dereferenceable(64) [[ARG:%.*]])
#1[[ATTR0]] {
; CGSCC-NEXT: bb:

```

```

; CGSCC-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT: [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT: call void @llvm.memset.p0.i64(ptr noalias nocapture nofree noundef nonnull writeonly align 64
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5]]
; CGSCC-NEXT: [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64
; CGSCC-NEXT: call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256(ptr
noalias nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64>
[[TMP0]]) #[[ATTR6]]
; CGSCC-NEXT:
    [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; CGSCC-NEXT: store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; CGSCC-NEXT: ret void
;
bb:
    %tmp = alloca <8 x i64>, align 32
    %tmp2 = alloca <8 x i64>, align 32
    call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
    call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256(ptr %tmp2, ptr %tmp)
    %tmp4 = load <8 x i64>, ptr %tmp2, align 32
    store <8 x i64> %tmp4, ptr %arg, align 2
    ret void
}

; This should promote
define internal fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512(ptr %arg, ptr
readonly %arg1) #0 {
;
; CHECK: Function Attrs: inlinehint mustprogress nofree norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CHECK-LABEL: define {{{^@}+}}@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512
; CHECK-SAME: (ptr noalias nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64)
[[ARG:%.*]],
    <8 x i64> [[TMP0:%.*]]) #[[ATTR0]] {
; CHECK-NEXT: bb:
; CHECK-NEXT: [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
; CHECK-NEXT: store <8 x i64> [[TMP0]], ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT: [[TMP:%.*]] = load <8 x i64>, ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT: store <8 x i64> [[TMP]], ptr [[ARG]], align 64
; CHECK-NEXT: ret void
;
bb:
    %tmp = load <8 x i64>, ptr %arg1
    store <8 x i64> %tmp, ptr %arg
    ret void
}

define void @avx512_legal512_prefer256_call_avx512_legal512_prefer512(ptr %arg) #1 {
;

```

```

; TUNIT: Function Attrs: inlinehint mustprogress norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; TUNIT-LABEL: define {{^[^@]+}}@avx512_legal512_prefer256_call_avx512_legal512_prefer512
; TUNIT-SAME: (ptr nocapture norecurse writeonly [[ARG:%.*]]) #[[ATTR1]] {
; TUNIT-NEXT: bb:
; TUNIT-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT:    [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT:    call void @llvm.memset.p0.i64(ptr
noalias nocapture norecurse noundef nonnull writeonly align 32 dereferenceable(64) [[TMP]], i8 noundef 0, i64
noundef 32, i1 noundef false) #[[ATTR5]]
; TUNIT-NEXT:    [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64
; TUNIT-NEXT:    call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512(ptr
noalias nocapture norecurse noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64>
[[TMP0]]) #[[ATTR6]]
; TUNIT-NEXT:    [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; TUNIT-NEXT:    store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; TUNIT-NEXT:    ret void
;
; CGSCC: Function Attrs: inlinehint mustprogress norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CGSCC-LABEL: define {{^[^@]+}}@avx512_legal512_prefer256_call_avx512_legal512_prefer512
; CGSCC-SAME: (ptr nocapture norecurse noundef nonnull writeonly align 2 dereferenceable(64) [[ARG:%.*]])
#[[ATTR1]] {
; CGSCC-NEXT: bb:
; CGSCC-NEXT:
    [[TMP:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT:    [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT:    call void @llvm.memset.p0.i64(ptr noalias nocapture norecurse noundef nonnull writeonly align 64
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5]]
; CGSCC-NEXT:    [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64
; CGSCC-NEXT:    call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512(ptr
noalias nocapture norecurse noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64>
[[TMP0]]) #[[ATTR6]]
; CGSCC-NEXT:    [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; CGSCC-NEXT:    store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; CGSCC-NEXT:    ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align 32
call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512(ptr
%tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

```

```

; This should not promote
define internal fastcc void @callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr %arg, ptr
readonly %arg1) #1 {
;
; CHECK: Function Attrs: inlinehint mustprogress nofree norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CHECK-LABEL: define {{[^@]+}}@callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256
; CHECK-SAME: (ptr noalias nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64)
[[ARG:%.*]], ptr noalias nocapture nofree noundef nonnull readonly align 64 dereferenceable(64) [[ARG1:%.*]])
#[[ATTR1]] {
; CHECK-NEXT: bb:
; CHECK-NEXT: [[TMP:%.*]] = load <8 x i64>, ptr [[ARG1]], align 64
; CHECK-NEXT: store <8 x i64> [[TMP]], ptr [[ARG]], align 64
; CHECK-NEXT: ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

define void @avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr %arg) #2 {
;
; TUNIT: Function Attrs: inlinehint mustprogress nofree norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; TUNIT-LABEL: define {{[^@]+}}@avx512_legal256_prefer256_call_avx512_legal512_prefer256
; TUNIT-SAME: (ptr nocapture nofree writeonly [[ARG:%.*]]) #[[ATTR2:[0-9]+]] {
; TUNIT-NEXT: bb:
; TUNIT-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT: [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT: call void @llvm.memset.p0.i64(ptr noalias nocapture nofree noundef nonnull writeonly align 32
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5]]
; TUNIT-NEXT: call fastcc void @callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr
noalias nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], ptr noalias
nocapture nofree noundef nonnull readonly align 64 dereferenceable(64)
[[TMP]]) #[[ATTR6]]
; TUNIT-NEXT: [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; TUNIT-NEXT: store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; TUNIT-NEXT: ret void
;
; CGSCC: Function Attrs: inlinehint mustprogress nofree norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CGSCC-LABEL: define {{[^@]+}}@avx512_legal256_prefer256_call_avx512_legal512_prefer256
; CGSCC-SAME: (ptr nocapture nofree noundef nonnull writeonly align 2 dereferenceable(64) [[ARG:%.*]])
#[[ATTR2:[0-9]+]] {
; CGSCC-NEXT: bb:
; CGSCC-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT: [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32

```



```

; CGSCC-NEXT:  call void @llvm.memset.p0.i64(ptr noalias nocapture norecurse noundef nonnull writeonly align 64
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5]]
; CGSCC-NEXT:  call fastcc void @callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr
noalias
nocapture norecurse noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], ptr noalias nocapture
norecurse noundef nonnull readonly align 64 dereferenceable(64) [[TMP]]) #[[ATTR6]]
; CGSCC-NEXT:  [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; CGSCC-NEXT:  store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; CGSCC-NEXT:  ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align 32
call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
call fastcc void @callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr %tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

; This should not promote
define internal fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr %arg, ptr
readonly %arg1) #2 {
;
; CHECK: Function Attrs: inlinehint mustprogress norecurse norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
;
CHECK-LABEL: define {{^[^@]+}}@callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256
; CHECK-SAME: (ptr noalias nocapture norecurse noundef nonnull writeonly align 64 dereferenceable(64)
[[ARG:%.*]], ptr noalias nocapture norecurse noundef nonnull readonly align 64 dereferenceable(64) [[ARG1:%.*]])
#[[ATTR2:[0-9]+]] {
; CHECK-NEXT: bb:
; CHECK-NEXT:  [[TMP:%.*]] = load <8 x i64>, ptr [[ARG1]], align 64
; CHECK-NEXT:  store <8 x i64> [[TMP]], ptr [[ARG]], align 64
; CHECK-NEXT:  ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

define void @avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr %arg) #1 {
;
; TUNIT: Function Attrs: inlinehint mustprogress norecurse norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; TUNIT-LABEL: define {{^[^@]+}}@avx512_legal512_prefer256_call_avx512_legal256_prefer256
; TUNIT-SAME: (ptr nocapture norecurse writeonly [[ARG:%.*]]) #[[ATTR1]] {

```

```

; TUNIT-NEXT: bb:
; TUNIT-NEXT:
    [[TMP:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT:  [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT:  call void @llvm.memset.p0.i64(ptr noalias nocapture nofree noundef nonnull writeonly align 32
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5]]
; TUNIT-NEXT:  call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr noalias
nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], ptr noalias
nocapture nofree noundef nonnull readonly align 64 dereferenceable(64) [[TMP]]) #[[ATTR6]]
; TUNIT-NEXT:  [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; TUNIT-NEXT:  store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; TUNIT-NEXT:  ret void
;
; CGSCC: Function Attrs: inlinehint mustprogress nofree norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CGSCC-LABEL: define {[@^@+]}@avx512_legal512_prefer256_call_avx512_legal256_prefer256
;
    CGSCC-SAME: (ptr nocapture nofree noundef nonnull writeonly align 2 dereferenceable(64) [[ARG:%.*]])
#[[ATTR1]] {
; CGSCC-NEXT: bb:
; CGSCC-NEXT:  [[TMP:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT:  [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT:  call void @llvm.memset.p0.i64(ptr noalias nocapture nofree noundef nonnull writeonly align 64
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5]]
; CGSCC-NEXT:  call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr noalias
nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], ptr noalias
nocapture nofree noundef nonnull readonly align 64 dereferenceable(64) [[TMP]]) #[[ATTR6]]
; CGSCC-NEXT:  [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; CGSCC-NEXT:  store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; CGSCC-NEXT:  ret void
;
bb:
    %tmp = alloca <8 x i64>, align 32
    %tmp2 = alloca <8 x i64>,
    align 32
    call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
    call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr %tmp2, ptr %tmp)
    %tmp4 = load <8 x i64>, ptr %tmp2, align 32
    store <8 x i64> %tmp4, ptr %arg, align 2
    ret void
}

; This should promote
define internal fastcc void @callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256(ptr %arg, ptr readonly
%arg1) #3 {
;
; CHECK: Function Attrs: inlinehint mustprogress nofree norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable

```

```

; CHECK-LABEL: define {{^[^@]+}}@callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256
; CHECK-SAME: (ptr noalias nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64)
[[ARG:%.*]], <8 x i64> [[TMP0:%.*]]) #[[ATTR3:[0-9]+]] {
; CHECK-NEXT: bb:
; CHECK-NEXT: [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
; CHECK-NEXT: store <8 x i64> [[TMP0]], ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT: [[TMP:%.*]]
= load <8 x i64>, ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT: store <8 x i64> [[TMP]], ptr [[ARG]], align 64
; CHECK-NEXT: ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

define void @avx2_legal256_prefer256_call_avx2_legal512_prefer256(ptr %arg) #4 {
;
; TUNIT: Function Attrs: inlinehint mustprogress nofree norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; TUNIT-LABEL: define {{^[^@]+}}@avx2_legal256_prefer256_call_avx2_legal512_prefer256
; TUNIT-SAME: (ptr nocapture nofree writeonly [[ARG:%.*]]) #[[ATTR3]] {
; TUNIT-NEXT: bb:
; TUNIT-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT: [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT: call void @llvm.memset.p0.i64(ptr noalias nocapture nofree noundef nonnull writeonly align 32
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5]]
; TUNIT-NEXT: [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64
;
; TUNIT-NEXT: call fastcc void @callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256(ptr noalias
nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64> [[TMP0]])
#[[ATTR6]]
; TUNIT-NEXT: [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; TUNIT-NEXT: store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; TUNIT-NEXT: ret void
;
; CGSCC: Function Attrs: inlinehint mustprogress nofree norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CGSCC-LABEL: define {{^[^@]+}}@avx2_legal256_prefer256_call_avx2_legal512_prefer256
; CGSCC-SAME: (ptr nocapture nofree noundef nonnull writeonly align 2 dereferenceable(64) [[ARG:%.*]])
#[[ATTR3]] {
; CGSCC-NEXT: bb:
; CGSCC-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT: [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT: call void @llvm.memset.p0.i64(ptr noalias nocapture nofree noundef nonnull writeonly align 64
dereferenceable(64) [[TMP]],

```

```

i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5]]
; CGSCC-NEXT:  [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64
; CGSCC-NEXT:  call fastcc void @callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256(ptr noalias
nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64> [[TMP0]])
#[[ATTR6]]
; CGSCC-NEXT:  [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; CGSCC-NEXT:  store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; CGSCC-NEXT:  ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align 32
call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
call fastcc void @callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256(ptr %tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

```

; This should promote

```

define internal fastcc void @callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256(ptr
%arg, ptr readonly %arg1) #4 {
;

```

```

; CHECK: Function Attrs: inlinehint mustprogress nofree norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable

```

```

; CHECK-LABEL: define {{^[^@]+}}@callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256
; CHECK-SAME: (ptr noalias nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64)
[[ARG:%.*]], <8 x i64> [[TMP0:%.*]]) #[[ATTR3]] {

```

```

; CHECK-NEXT: bb:

```

```

; CHECK-NEXT:  [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
; CHECK-NEXT:  store <8 x i64> [[TMP0]], ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT:  [[TMP:%.*]] = load <8 x i64>, ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT:  store <8 x i64> [[TMP]], ptr [[ARG]], align 64
; CHECK-NEXT:  ret void
;

```

```

bb:

```

```

%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

```

```

define void @avx2_legal512_prefer256_call_avx2_legal256_prefer256(ptr %arg) #3 {
;

```

```

; TUNIT: Function Attrs: inlinehint mustprogress nofree norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable

```

```

; TUNIT-LABEL: define {{^[^@]+}}@avx2_legal512_prefer256_call_avx2_legal256_prefer256
; TUNIT-SAME: (ptr nocapture nofree writeonly [[ARG:%.*]]) #[[ATTR3]] {

```

```

; TUNIT-NEXT: bb:

```

```

; TUNIT-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT: [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT: call void @llvm.memset.p0.i64(ptr noalias nocapture nofree noundef nonnull writeonly align 32
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5]]
; TUNIT-NEXT: [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64
; TUNIT-NEXT: call fastcc void @callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256(ptr noalias
nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64> [[TMP0]])
#[[ATTR6]]
; TUNIT-NEXT: [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; TUNIT-NEXT: store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; TUNIT-NEXT: ret void
;
; CGSCC: Function
Attrs: inlinehint mustprogress nofree norecurse nosync nounwind willreturn memory(argmem: readwrite) uwtable
; CGSCC-LABEL: define { {^[@]}+ } @avx2_legal512_prefer256_call_avx2_legal256_prefer256
; CGSCC-SAME: (ptr nocapture nofree noundef nonnull writeonly align 2 dereferenceable(64) [[ARG:%.*]])
#[[ATTR3]] {
; CGSCC-NEXT: bb:
; CGSCC-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT: [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT: call void @llvm.memset.p0.i64(ptr noalias nocapture nofree noundef nonnull writeonly align 64
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5]]
; CGSCC-NEXT: [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64
; CGSCC-NEXT: call fastcc void @callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256(ptr noalias
nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64> [[TMP0]])
#[[ATTR6]]
; CGSCC-NEXT: [[TRUETMP4:%.*]] = load <8 x i64>,
ptr [[TRUETMP2]], align 64
; CGSCC-NEXT: store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; CGSCC-NEXT: ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align 32
call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
call fastcc void @callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256(ptr %tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

; Function Attrs: argmemonly nounwind
declare void @llvm.memset.p0.i64(ptr nocapture writeonly, i8, i64, i1) #5

attributes #0 = { inlinehint norecurse nounwind uwtable "target-features"="+avx512vl" "min-legal-vector-
width"="512" "prefer-vector-width"="512" }
attributes #1 = { inlinehint norecurse nounwind uwtable "target-features"="+avx512vl" "min-legal-vector-
width"="512" "prefer-vector-width"="256" }

```

```

attributes #2 = { inlinehint norecurse nounwind uwtable "target-features"="+avx512vl" "min-legal-vector-
width"="256"
"prefer-vector-width"="256" }
attributes #3 = { inlinehint norecurse nounwind uwtable "target-features"="+avx2" "min-legal-vector-width"="512"
"prefer-vector-width"="256" }
attributes #4 = { inlinehint norecurse nounwind uwtable "target-features"="+avx2" "min-legal-vector-width"="256"
"prefer-vector-width"="256" }
attributes #5 = { argmemonly nounwind }
;
; TUNIT: attributes #[[ATTR0]] = { inlinehint mustprogress nofree norecurse nosync nounwind willreturn
memory(argmem: readwrite) uwtable "min-legal-vector-width"="512" "prefer-vector-width"="512" "target-
features"="+avx512vl" }
; TUNIT: attributes #[[ATTR1]] = { inlinehint mustprogress nofree norecurse nosync nounwind willreturn
memory(argmem: readwrite) uwtable "min-legal-vector-width"="512" "prefer-vector-width"="256" "target-
features"="+avx512vl" }
; TUNIT: attributes #[[ATTR2]] = { inlinehint mustprogress nofree norecurse nosync nounwind willreturn
memory(argmem: readwrite) uwtable "min-legal-vector-width"="256" "prefer-vector-width"="256"
"target-features"="+avx512vl" }
; TUNIT: attributes #[[ATTR3]] = { inlinehint mustprogress nofree norecurse nosync nounwind willreturn
memory(argmem: readwrite) uwtable "min-legal-vector-width"="512" "prefer-vector-width"="256" "target-
features"="+avx2" }
; TUNIT: attributes #[[ATTR4:[0-9]+]] = { nocallback nofree nounwind willreturn memory(argmem: write) }
; TUNIT: attributes #[[ATTR5]] = { nofree willreturn memory(write) }
; TUNIT: attributes #[[ATTR6]] = { nofree nosync nounwind willreturn }
;
; CGSCC: attributes #[[ATTR0]] = { inlinehint mustprogress nofree norecurse nosync nounwind willreturn
memory(argmem: readwrite) uwtable "min-legal-vector-width"="512" "prefer-vector-width"="512" "target-
features"="+avx512vl" }
; CGSCC: attributes #[[ATTR1]] = { inlinehint mustprogress nofree norecurse nosync nounwind willreturn
memory(argmem: readwrite) uwtable "min-legal-vector-width"="512" "prefer-vector-width"="256" "target-
features"="+avx512vl" }
; CGSCC: attributes #[[ATTR2]]
= { inlinehint mustprogress nofree norecurse nosync nounwind willreturn memory(argmem: readwrite) uwtable
"min-legal-vector-width"="256" "prefer-vector-width"="256" "target-features"="+avx512vl" }
; CGSCC: attributes #[[ATTR3]] = { inlinehint mustprogress nofree norecurse nosync nounwind willreturn
memory(argmem: readwrite) uwtable "min-legal-vector-width"="512" "prefer-vector-width"="256" "target-
features"="+avx2" }
; CGSCC: attributes #[[ATTR4:[0-9]+]] = { nocallback nofree nounwind willreturn memory(argmem: write) }
; CGSCC: attributes #[[ATTR5]] = { nofree willreturn memory(write) }
; CGSCC: attributes #[[ATTR6]] = { nofree nounwind willreturn }
;
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64I
; RUN: llc -mtriple=riscv64 -mattr=+zbb -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64ZBB

declare i32 @llvm.ctlz.i32(i32, i1)

```

```

define signext i32 @ctlz_i32(i32 signext %a) nounwind {
; RV64I-LABEL: ctlz_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT: beqz a0, .LBB0_2
; RV64I-NEXT: # %bb.1: # %cond.false
; RV64I-NEXT: addi sp, sp, -16
; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT: srliw a1, a0, 1
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 2
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 4
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 8
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 16
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT:
    not a0, a0
; RV64I-NEXT: srliw a1, a0, 1
; RV64I-NEXT: lui a2, 349525
; RV64I-NEXT: addi a2, a2, 1365
; RV64I-NEXT: and a1, a1, a2
; RV64I-NEXT: subw a0, a0, a1
; RV64I-NEXT: lui a1, 209715
; RV64I-NEXT: addi a1, a1, 819
; RV64I-NEXT: and a2, a0, a1
; RV64I-NEXT: srliw a0, a0, 2
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: add a0, a2, a0
; RV64I-NEXT: srliw a1, a0, 4
; RV64I-NEXT: addw a0, a0, a1
; RV64I-NEXT: lui a1, 61681
; RV64I-NEXT: addiw a1, a1, -241
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: lui a1, 4112
; RV64I-NEXT: addiw a1, a1, 257
; RV64I-NEXT: call __muldi3
; RV64I-NEXT: srliw a0, a0, 24
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT: addi sp, sp, 16
; RV64I-NEXT: ret
; RV64I-NEXT: .LBB0_2:
; RV64I-NEXT: li a0, 32
; RV64I-NEXT: ret
;
; RV64ZBB-LABEL: ctlz_i32:

```

```

; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT:  clzw a0, a0
; RV64ZBB-NEXT:  ret
%1 = call i32 @llvmctlz.i32(i32 %a, i1 false)
ret i32 %1
}

define signext i32 @log2_i32(i32 signext %a) nounwind {
; RV64I-LABEL: log2_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT:  beqz a0, .LBB1_2
; RV64I-NEXT:  # %bb.1: # %cond.false
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  srliw a1, a0, 1
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 2
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 4
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 8
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 16
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  srliw a1, a0, 1
; RV64I-NEXT:  lui a2, 349525
; RV64I-NEXT:  addi a2, a2, 1365
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  subw a0, a0, a1
; RV64I-NEXT:  lui a1, 209715
; RV64I-NEXT:  addi a1, a1, 819
; RV64I-NEXT:  and a2, a0, a1
; RV64I-NEXT:  srliw a0, a0, 2
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  add a0, a2, a0
; RV64I-NEXT:
    srliw a1, a0, 4
; RV64I-NEXT:  addw a0, a0, a1
; RV64I-NEXT:  lui a1, 61681
; RV64I-NEXT:  addiw a1, a1, -241
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  lui a1, 4112
; RV64I-NEXT:  addiw a1, a1, 257
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:  srliw a0, a0, 24
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16

```



```

; RV64I-NEXT:  j .LBB1_3
; RV64I-NEXT:  .LBB1_2:
; RV64I-NEXT:  li a0, 32
; RV64I-NEXT:  .LBB1_3: # %cond.end
; RV64I-NEXT:  li a1, 31
; RV64I-NEXT:  subw a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: log2_i32:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  clzw a0, a0
; RV64ZBB-NEXT:  li a1, 31
; RV64ZBB-NEXT:  subw a0, a1, a0
; RV64ZBB-NEXT:  ret
%1 = call i32 @llvmctlz.i32(i32 %a, i1 false)
%2 = sub i32 31, %1
ret i32 %2
}

define signext i32 @log2_ceil_i32(i32 signext %a) nounwind {
; RV64I-LABEL: log2_ceil_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra,
8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  sd s0, 0(sp) # 8-byte Folded Spill
; RV64I-NEXT:  addiw a0, a0, -1
; RV64I-NEXT:  li s0, 32
; RV64I-NEXT:  li a1, 32
; RV64I-NEXT:  beqz a0, .LBB2_2
; RV64I-NEXT:  # %bb.1: # %cond.false
; RV64I-NEXT:  srliw a1, a0, 1
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 2
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 4
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 8
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 16
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  srliw a1, a0, 1
; RV64I-NEXT:  lui a2, 349525
; RV64I-NEXT:  addi a2, a2, 1365
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  subw a0, a0, a1
; RV64I-NEXT:  lui a1, 209715

```

```

; RV64I-NEXT:  addi a1, a1, 819
; RV64I-NEXT:  and a2, a0, a1
; RV64I-NEXT:  srliw a0, a0, 2
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  add a0, a2, a0
; RV64I-NEXT:  srliw a1, a0, 4
; RV64I-NEXT:
    addw a0, a0, a1
; RV64I-NEXT:  lui a1, 61681
; RV64I-NEXT:  addiw a1, a1, -241
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  lui a1, 4112
; RV64I-NEXT:  addiw a1, a1, 257
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:  srliw a1, a0, 24
; RV64I-NEXT:  .LBB2_2: # %cond.end
; RV64I-NEXT:  subw a0, s0, a1
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  ld s0, 0(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: log2_ceil_i32:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  addi a0, a0, -1
; RV64ZBB-NEXT:  clzw a0, a0
; RV64ZBB-NEXT:  li a1, 32
; RV64ZBB-NEXT:  subw a0, a1, a0
; RV64ZBB-NEXT:  ret
%1 = sub i32 %a, 1
%2 = call i32 @llvmctlz.i32(i32 %1, i1 false)
%3 = sub i32 32, %2
ret i32 %3
}

define signext i32 @findLastSet_i32(i32 signext %a) nounwind {
; RV64I-LABEL: findLastSet_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
;
    RV64I-NEXT:  sd s0, 0(sp) # 8-byte Folded Spill
; RV64I-NEXT:  mv s0, a0
; RV64I-NEXT:  srliw a0, a0, 1
; RV64I-NEXT:  or a0, s0, a0
; RV64I-NEXT:  srliw a1, a0, 2
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 4

```

```

; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 8
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 16
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: not a0, a0
; RV64I-NEXT: srliw a1, a0, 1
; RV64I-NEXT: lui a2, 349525
; RV64I-NEXT: addi a2, a2, 1365
; RV64I-NEXT: and a1, a1, a2
; RV64I-NEXT: subw a0, a0, a1
; RV64I-NEXT: lui a1, 209715
; RV64I-NEXT: addi a1, a1, 819
; RV64I-NEXT: and a2, a0, a1
; RV64I-NEXT: srliw a0, a0, 2
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: add a0, a2, a0
; RV64I-NEXT: srliw a1, a0, 4
; RV64I-NEXT: addw a0, a0, a1
; RV64I-NEXT: lui a1, 61681
; RV64I-NEXT: addiw a1, a1, -241
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: lui a1, 4112
; RV64I-NEXT:
    addiw a1, a1, 257
; RV64I-NEXT: call __muldi3
; RV64I-NEXT: srliw a0, a0, 24
; RV64I-NEXT: xori a0, a0, 31
; RV64I-NEXT: snez a1, s0
; RV64I-NEXT: addiw a1, a1, -1
; RV64I-NEXT: or a0, a1, a0
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT: ld s0, 0(sp) # 8-byte Folded Reload
; RV64I-NEXT: addi sp, sp, 16
; RV64I-NEXT: ret
;
; RV64ZBB-LABEL: findLastSet_i32:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT: clzw a1, a0
; RV64ZBB-NEXT: xori a1, a1, 31
; RV64ZBB-NEXT: snez a0, a0
; RV64ZBB-NEXT: addiw a0, a0, -1
; RV64ZBB-NEXT: or a0, a0, a1
; RV64ZBB-NEXT: ret
%1 = call i32 @llvm.ctlz.i32(i32 %a, i1 true)
%2 = xor i32 31, %1
%3 = icmp eq i32 %a, 0
%4 = select i1 %3, i32 -1, i32 %2

```

```
ret i32 %4
}
```

```
define i32 @ctlz_lshr_i32(i32 signext %a) {
; RV64I-LABEL: ctlz_lshr_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT: srliw a0, a0, 1
; RV64I-NEXT: beqz a0, .LBB4_2
; RV64I-NEXT: # %bb.1: # %cond.false
; RV64I-NEXT: addi sp, sp,
-16
; RV64I-NEXT: .cfi_def_cfa_offset 16
; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT: .cfi_offset ra, -8
; RV64I-NEXT: srliw a1, a0, 1
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 2
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 4
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 8
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 16
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: not a0, a0
; RV64I-NEXT: srliw a1, a0, 1
; RV64I-NEXT: lui a2, 349525
; RV64I-NEXT: addi a2, a2, 1365
; RV64I-NEXT: and a1, a1, a2
; RV64I-NEXT: subw a0, a0, a1
; RV64I-NEXT: lui a1, 209715
; RV64I-NEXT: addi a1, a1, 819
; RV64I-NEXT: and a2, a0, a1
; RV64I-NEXT: srliw a0, a0, 2
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: add a0, a2, a0
; RV64I-NEXT: srliw a1, a0, 4
; RV64I-NEXT: addw a0, a0, a1
; RV64I-NEXT: lui a1, 61681
; RV64I-NEXT: addiw a1, a1, -241
; RV64I-NEXT: and a0,
a0, a1
; RV64I-NEXT: lui a1, 4112
; RV64I-NEXT: addiw a1, a1, 257
; RV64I-NEXT: call __muldi3
; RV64I-NEXT: srliw a0, a0, 24
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT: addi sp, sp, 16
```

```

; RV64I-NEXT:  ret
; RV64I-NEXT:  .LBB4_2:
; RV64I-NEXT:  li a0, 32
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: ctlz_lshr_i32:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  srliw a0, a0, 1
; RV64ZBB-NEXT:  clzw a0, a0
; RV64ZBB-NEXT:  ret
%1 = lshr i32 %a, 1
%2 = call i32 @llvm.ctlz.i32(i32 %1, i1 false)
ret i32 %2
}

declare i64 @llvm.ctlz.i64(i64, i1)

define i64 @ctlz_i64(i64 %a) nounwind {
; RV64I-LABEL: ctlz_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  beqz a0, .LBB5_2
; RV64I-NEXT:  # %bb.1: # %cond.false
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  srli a1, a0, 1
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srli a1, a0, 2
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srli a1, a0, 4
; RV64I-NEXT:
    or a0, a0, a1
; RV64I-NEXT:  srli a1, a0, 8
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srli a1, a0, 16
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srli a1, a0, 32
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  srli a1, a0, 1
; RV64I-NEXT:  lui a2, 349525
; RV64I-NEXT:  addiw a2, a2, 1365
; RV64I-NEXT:  slli a3, a2, 32
; RV64I-NEXT:  add a2, a2, a3
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  sub a0, a0, a1
; RV64I-NEXT:  lui a1, 209715
; RV64I-NEXT:  addiw a1, a1, 819
; RV64I-NEXT:  slli a2, a1, 32

```

```

; RV64I-NEXT:  add a1, a1, a2
; RV64I-NEXT:  and a2, a0, a1
; RV64I-NEXT:  srli a0, a0, 2
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  add a0, a2, a0
; RV64I-NEXT:  srli a1, a0, 4
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  lui a1, 61681
; RV64I-NEXT:  addiw a1, a1, -241
; RV64I-NEXT:  slli a2, a1, 32
; RV64I-NEXT:  add a1, a1, a2
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  lui a1, 4112
; RV64I-NEXT:
    addiw a1, a1, 257
; RV64I-NEXT:  slli a2, a1, 32
; RV64I-NEXT:  add a1, a1, a2
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:  srli a0, a0, 56
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
; RV64I-NEXT: .LBB5_2:
; RV64I-NEXT:  li a0, 64
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: ctlz_i64:
; RV64ZBB:     # %bb.0:
; RV64ZBB-NEXT:  clz a0, a0
; RV64ZBB-NEXT:  ret
%1 = call i64 @llvm.ctlz.i64(i64 %a, i1 false)
ret i64 %1
}

```

```

declare i32 @llvm.cttz.i32(i32, i1)

```

```

define signext i32 @cttz_i32(i32 signext %a) nounwind {
; RV64I-LABEL: cttz_i32:
; RV64I:     # %bb.0:
; RV64I-NEXT:  beqz a0, .LBB6_2
; RV64I-NEXT: # %bb.1: # %cond.false
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  negw a1, a0
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  lui a1, 30667
; RV64I-NEXT:  addiw a1, a1, 1329
; RV64I-NEXT:  call __muldi3

```

```

; RV64I-NEXT:  srlw a0, a0, 27
;
RV64I-NEXT:  lui a1, %hi(.LCPI6_0)
; RV64I-NEXT:  addi a1, a1, %lo(.LCPI6_0)
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  lbu a0, 0(a0)
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
; RV64I-NEXT:  .LBB6_2:
; RV64I-NEXT:  li a0, 32
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: cttz_i32:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  ctzw a0, a0
; RV64ZBB-NEXT:  ret
%1 = call i32 @llvm.cttz.i32(i32 %a, i1 false)
ret i32 %1
}

define signext i32 @cttz_zero_undef_i32(i32 signext %a) nounwind {
; RV64I-LABEL: cttz_zero_undef_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  negw a1, a0
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  lui a1, 30667
; RV64I-NEXT:  addiw a1, a1, 1329
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:  srlw a0, a0, 27
; RV64I-NEXT:  lui a1, %hi(.LCPI7_0)
; RV64I-NEXT:  addi a1, a1, %lo(.LCPI7_0)
; RV64I-NEXT:
    add a0, a1, a0
; RV64I-NEXT:  lbu a0, 0(a0)
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: cttz_zero_undef_i32:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  ctzw a0, a0
; RV64ZBB-NEXT:  ret
%1 = call i32 @llvm.cttz.i32(i32 %a, i1 true)
ret i32 %1
}

```

```

define signext i32 @findFirstSet_i32(i32 signext %a) nounwind {
; RV64I-LABEL: findFirstSet_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  sd s0, 0(sp) # 8-byte Folded Spill
; RV64I-NEXT:  mv s0, a0
; RV64I-NEXT:  negw a0, a0
; RV64I-NEXT:  and a0, s0, a0
; RV64I-NEXT:  lui a1, 30667
; RV64I-NEXT:  addiw a1, a1, 1329
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:  srlw a0, a0, 27
; RV64I-NEXT:  lui a1, %hi(.LCPI8_0)
; RV64I-NEXT:  addi a1, a1, %lo(.LCPI8_0)
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  lbu a0, 0(a0)
; RV64I-NEXT:  snez a1, s0
; RV64I-NEXT:
    addi a1, a1, -1
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  ld s0, 0(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: findFirstSet_i32:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  ctw a1, a0
; RV64ZBB-NEXT:  snez a0, a0
; RV64ZBB-NEXT:  addiw a0, a0, -1
; RV64ZBB-NEXT:  or a0, a0, a1
; RV64ZBB-NEXT:  ret
%1 = call i32 @llvm.cttz.i32(i32 %a, i1 true)
%2 = icmp eq i32 %a, 0
%3 = select i1 %2, i32 -1, i32 %1
ret i32 %3
}

```

```

define signext i32 @ffs_i32(i32 signext %a) nounwind {
; RV64I-LABEL: ffs_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  sd s0, 0(sp) # 8-byte Folded Spill
; RV64I-NEXT:  mv s0, a0
; RV64I-NEXT:  negw a0, a0

```



```

; RV64I-NEXT:  and a0, s0, a0
; RV64I-NEXT:  lui a1, 30667
; RV64I-NEXT:  addiw a1, a1, 1329
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:
    srlw a0, a0, 27
; RV64I-NEXT:  lui a1, %hi(.LCPI9_0)
; RV64I-NEXT:  addi a1, a1, %lo(.LCPI9_0)
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  lbu a0, 0(a0)
; RV64I-NEXT:  addi a0, a0, 1
; RV64I-NEXT:  seqz a1, s0
; RV64I-NEXT:  addi a1, a1, -1
; RV64I-NEXT:  and a0, a1, a0
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  ld s0, 0(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: ffs_i32:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  ctzw a1, a0
; RV64ZBB-NEXT:  addi a1, a1, 1
; RV64ZBB-NEXT:  seqz a0, a0
; RV64ZBB-NEXT:  addi a0, a0, -1
; RV64ZBB-NEXT:  and a0, a0, a1
; RV64ZBB-NEXT:  zext.h a0, a0
; RV64ZBB-NEXT:  ret
%1 = call i32 @llvm.cttz.i32(i32 %a, i1 true)
%2 = add i32 %1, 1
%3 = icmp eq i32 %a, 0
%4 = select i1 %3, i32 0, i32 %2
ret i32 %4
}

declare i64 @llvm.cttz.i64(i64, i1)

define i64 @cttz_i64(i64
    %a) nounwind {
; RV64I-LABEL: cttz_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  beqz a0, .LBB10_2
; RV64I-NEXT:  # %bb.1: # %cond.false
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  neg a1, a0

```

```

; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: lui a1, %hi(.LCPI10_0)
; RV64I-NEXT: ld a1, %lo(.LCPI10_0)(a1)
; RV64I-NEXT: call __muldi3
; RV64I-NEXT: srli a0, a0, 58
; RV64I-NEXT: lui a1, %hi(.LCPI10_1)
; RV64I-NEXT: addi a1, a1, %lo(.LCPI10_1)
; RV64I-NEXT: add a0, a1, a0
; RV64I-NEXT: lbu a0, 0(a0)
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT: addi sp, sp, 16
; RV64I-NEXT: ret
; RV64I-NEXT: .LBB10_2:
; RV64I-NEXT: li a0, 64
; RV64I-NEXT: ret
;
; RV64ZBB-LABEL: cttz_i64:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT: ctz a0, a0
; RV64ZBB-NEXT: ret
%1 = call i64 @llvm.cttz.i64(i64 %a, i1 false)
ret i64 %1
}

```

```
declare i32 @llvm.ctpop.i32(i32)
```

```

define signext i32 @ctpop_i32(i32
signext %a) nounwind {
; RV64I-LABEL: ctpop_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT: addi sp, sp, -16
; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT: srliw a1, a0, 1
; RV64I-NEXT: lui a2, 349525
; RV64I-NEXT: addi a2, a2, 1365
; RV64I-NEXT: and a1, a1, a2
; RV64I-NEXT: subw a0, a0, a1
; RV64I-NEXT: lui a1, 209715
; RV64I-NEXT: addi a1, a1, 819
; RV64I-NEXT: and a2, a0, a1
; RV64I-NEXT: srliw a0, a0, 2
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: add a0, a2, a0
; RV64I-NEXT: srliw a1, a0, 4
; RV64I-NEXT: addw a0, a0, a1
; RV64I-NEXT: lui a1, 61681
; RV64I-NEXT: addiw a1, a1, -241
; RV64I-NEXT: and a0, a0, a1

```

```

; RV64I-NEXT:  lui a1, 4112
; RV64I-NEXT:  addiw a1, a1, 257
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:  srlw a0, a0, 24
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: ctpop_i32:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  cpopw
a0, a0
; RV64ZBB-NEXT:  ret
%1 = call i32 @llvm.ctpop.i32(i32 %a)
ret i32 %1
}

```

```

define signext i32 @ctpop_i32_load(ptr %p) nounwind {
; RV64I-LABEL: ctpop_i32_load:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  lw a0, 0(a0)
; RV64I-NEXT:  srlw a1, a0, 1
; RV64I-NEXT:  lui a2, 349525
; RV64I-NEXT:  addi a2, a2, 1365
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  subw a0, a0, a1
; RV64I-NEXT:  lui a1, 209715
; RV64I-NEXT:  addi a1, a1, 819
; RV64I-NEXT:  and a2, a0, a1
; RV64I-NEXT:  srlw a0, a0, 2
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  add a0, a2, a0
; RV64I-NEXT:  srlw a1, a0, 4
; RV64I-NEXT:  addw a0, a0, a1
; RV64I-NEXT:  lui a1, 61681
; RV64I-NEXT:  addiw a1, a1, -241
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  lui a1, 4112
; RV64I-NEXT:  addiw a1, a1, 257
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:  srlw a0, a0, 24
; RV64I-NEXT:  ld ra, 8(sp)
# 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;

```

```

; RV64ZBB-LABEL: ctpop_i32_load:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT: lw a0, 0(a0)
; RV64ZBB-NEXT: cpopw a0, a0
; RV64ZBB-NEXT: ret
%a = load i32, ptr %p
%1 = call i32 @llvm.ctpop.i32(i32 %a)
ret i32 %1
}

```

```

declare i64 @llvm.ctpop.i64(i64)

```

```

define i64 @ctpop_i64(i64 %a) nounwind {
; RV64I-LABEL: ctpop_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT: addi sp, sp, -16
; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT: srli a1, a0, 1
; RV64I-NEXT: lui a2, 349525
; RV64I-NEXT: addiw a2, a2, 1365
; RV64I-NEXT: slli a3, a2, 32
; RV64I-NEXT: add a2, a2, a3
; RV64I-NEXT: and a1, a1, a2
; RV64I-NEXT: sub a0, a0, a1
; RV64I-NEXT: lui a1, 209715
; RV64I-NEXT: addiw a1, a1, 819
; RV64I-NEXT: slli a2, a1, 32
; RV64I-NEXT: add a1, a1, a2
; RV64I-NEXT: and a2, a0, a1
; RV64I-NEXT: srli a0, a0, 2
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: add a0, a2,
a0
; RV64I-NEXT: srli a1, a0, 4
; RV64I-NEXT: add a0, a0, a1
; RV64I-NEXT: lui a1, 61681
; RV64I-NEXT: addiw a1, a1, -241
; RV64I-NEXT: slli a2, a1, 32
; RV64I-NEXT: add a1, a1, a2
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: lui a1, 4112
; RV64I-NEXT: addiw a1, a1, 257
; RV64I-NEXT: slli a2, a1, 32
; RV64I-NEXT: add a1, a1, a2
; RV64I-NEXT: call __muldi3
; RV64I-NEXT: srli a0, a0, 56
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT: addi sp, sp, 16

```

```

; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: ctpop_i64:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  cpop a0, a0
; RV64ZBB-NEXT:  ret
%1 = call i64 @llvm.ctpop.i64(i64 %a)
ret i64 %1
}

```

```

define signext i32 @sextb_i32(i32 signext %a) nounwind {
; RV64I-LABEL: sextb_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 56
; RV64I-NEXT:  srai a0, a0, 56
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: sextb_i32:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  sext.b a0, a0
; RV64ZBB-NEXT:
    ret
%shl = shl i32 %a, 24
%shr = ashr exact i32 %shl, 24
ret i32 %shr
}

```

```

define i64 @sextb_i64(i64 %a) nounwind {
; RV64I-LABEL: sextb_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 56
; RV64I-NEXT:  srai a0, a0, 56
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: sextb_i64:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  sext.b a0, a0
; RV64ZBB-NEXT:  ret
%shl = shl i64 %a, 56
%shr = ashr exact i64 %shl, 56
ret i64 %shr
}

```

```

define signext i32 @sextb_i32(i32 signext %a) nounwind {
; RV64I-LABEL: sextb_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srai a0, a0, 48

```

```

; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: sixth_i32:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  sext.h a0, a0
; RV64ZBB-NEXT:  ret
%shl = shl i32 %a, 16
%shr = ashr exact i32 %shl, 16
ret i32 %shr
}

define i64 @sixth_i64(i64 %a) nounwind {
; RV64I-LABEL: sixth_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srai a0, a0, 48
; RV64I-NEXT:
    ret
;
; RV64ZBB-LABEL: sixth_i64:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  sext.h a0, a0
; RV64ZBB-NEXT:  ret
%shl = shl i64 %a, 48
%shr = ashr exact i64 %shl, 48
ret i64 %shr
}

define signext i32 @min_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: min_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  blt a0, a1, .LBB18_2
; RV64I-NEXT:  # %bb.1:
; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT:  .LBB18_2:
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: min_i32:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  min a0, a0, a1
; RV64ZBB-NEXT:  ret
%cmp = icmp slt i32 %a, %b
%cond = select i1 %cmp, i32 %a, i32 %b
ret i32 %cond
}

define i64 @min_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: min_i64:

```

```

; RV64I:    # %bb.0:
; RV64I-NEXT:  blt a0, a1, .LBB19_2
; RV64I-NEXT: # %bb.1:
; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT: .LBB19_2:
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: min_i64:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT:  min a0, a0, a1
; RV64ZBB-NEXT:  ret
%cmp = icmp
slt i64 %a, %b
%cond = select i1 %cmp, i64 %a, i64 %b
ret i64 %cond
}

define signext i32 @max_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: max_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT:  blt a1, a0, .LBB20_2
; RV64I-NEXT: # %bb.1:
; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT: .LBB20_2:
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: max_i32:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT:  max a0, a0, a1
; RV64ZBB-NEXT:  ret
%cmp = icmp sgt i32 %a, %b
%cond = select i1 %cmp, i32 %a, i32 %b
ret i32 %cond
}

define i64 @max_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: max_i64:
; RV64I:    # %bb.0:
; RV64I-NEXT:  blt a1, a0, .LBB21_2
; RV64I-NEXT: # %bb.1:
; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT: .LBB21_2:
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: max_i64:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT:  max a0, a0, a1
; RV64ZBB-NEXT:  ret

```

```

%cmp = icmp sgt i64 %a, %b
%cond = select i1 %cmp, i64 %a, i64 %b
ret i64 %cond
}

```

```

define signext i32 @minu_i32(i32 signext
%a, i32 signext %b) nounwind {
; RV64I-LABEL: minu_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  bltu a0, a1, .LBB22_2
; RV64I-NEXT: # %bb.1:
; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT: .LBB22_2:
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: minu_i32:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT:  minu a0, a0, a1
; RV64ZBB-NEXT:  ret
%cmp = icmp ult i32 %a, %b
%cond = select i1 %cmp, i32 %a, i32 %b
ret i32 %cond
}

```

```

define i64 @minu_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: minu_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  bltu a0, a1, .LBB23_2
; RV64I-NEXT: # %bb.1:
; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT: .LBB23_2:
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: minu_i64:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT:  minu a0, a0, a1
; RV64ZBB-NEXT:  ret
%cmp = icmp ult i64 %a, %b
%cond = select i1 %cmp, i64 %a, i64 %b
ret i64 %cond
}

```

```

define signext i32 @maxu_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: maxu_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  bltu a1,
a0, .LBB24_2
; RV64I-NEXT: # %bb.1:

```



```

; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT:  .LBB24_2:
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: maxu_i32:
; RV64ZBB:     # %bb.0:
; RV64ZBB-NEXT:  maxu a0, a0, a1
; RV64ZBB-NEXT:  ret
%cmp = icmp ugt i32 %a, %b
%cond = select i1 %cmp, i32 %a, i32 %b
ret i32 %cond
}

```

```

define i64 @maxu_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: maxu_i64:
; RV64I:       # %bb.0:
; RV64I-NEXT:  bltu a1, a0, .LBB25_2
; RV64I-NEXT:  # %bb.1:
; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT:  .LBB25_2:
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: maxu_i64:
; RV64ZBB:     # %bb.0:
; RV64ZBB-NEXT:  maxu a0, a0, a1
; RV64ZBB-NEXT:  ret
%cmp = icmp ugt i64 %a, %b
%cond = select i1 %cmp, i64 %a, i64 %b
ret i64 %cond
}

```

```

declare i32 @llvm.abs.i32(i32, i1 immarg)

```

```

define i32 @abs_i32(i32 %x) {
; RV64I-LABEL: abs_i32:
; RV64I:       # %bb.0:
; RV64I-NEXT:  sraiw a1, a0, 31
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  subw a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL:
abs_i32:
; RV64ZBB:     # %bb.0:
; RV64ZBB-NEXT:  sraiw a1, a0, 31
; RV64ZBB-NEXT:  xor a0, a0, a1
; RV64ZBB-NEXT:  subw a0, a0, a1
; RV64ZBB-NEXT:  ret

```

```

%abs = tail call i32 @llvm.abs.i32(i32 %x, i1 true)
ret i32 %abs
}

```

```

define signext i32 @abs_i32_sext(i32 signext %x) {
; RV64I-LABEL: abs_i32_sext:
; RV64I:      # %bb.0:
; RV64I-NEXT:  sraiw a1, a0, 31
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  subw a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: abs_i32_sext:
; RV64ZBB:     # %bb.0:
; RV64ZBB-NEXT:  sraiw a1, a0, 31
; RV64ZBB-NEXT:  xor a0, a0, a1
; RV64ZBB-NEXT:  subw a0, a0, a1
; RV64ZBB-NEXT:  ret
%abs = tail call i32 @llvm.abs.i32(i32 %x, i1 true)
ret i32 %abs
}

```

```

declare i64 @llvm.abs.i64(i64, i1 immarg)

```

```

define i64 @abs_i64(i64 %x) {
; RV64I-LABEL: abs_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srai a1, a0, 63
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  sub a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: abs_i64:
; RV64ZBB:
; RV64ZBB:     # %bb.0:
; RV64ZBB-NEXT:  neg a1, a0
; RV64ZBB-NEXT:  max a0, a0, a1
; RV64ZBB-NEXT:  ret
%abs = tail call i64 @llvm.abs.i64(i64 %x, i1 true)
ret i64 %abs
}

```

```

define i32 @zexth_i32(i32 %a) nounwind {
; RV64I-LABEL: zexth_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srli a0, a0, 48
; RV64I-NEXT:  ret

```

```

;
; RV64ZBB-LABEL: zexth_i32:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT:  slli a0, a0, 48
; RV64ZBB-NEXT:  srli a0, a0, 48
; RV64ZBB-NEXT:  ret
%and = and i32 %a, 65535
ret i32 %and
}

define i64 @zexth_i64(i64 %a) nounwind {
; RV64I-LABEL: zexth_i64:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srli a0, a0, 48
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: zexth_i64:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT:  zext.h a0, a0
; RV64ZBB-NEXT:  ret
%and = and i64 %a, 65535
ret i64 %and
}

declare i32 @llvm.bswap.i32(i32)

define signext i32 @bswap_i32(i32 signext %a) nounwind {
; RV64I-LABEL: bswap_i32:
;
; RV64I:    # %bb.0:
; RV64I-NEXT:  srliw a1, a0, 8
; RV64I-NEXT:  lui a2, 16
; RV64I-NEXT:  addiw a2, a2, -256
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  srliw a3, a0, 24
; RV64I-NEXT:  or a1, a1, a3
; RV64I-NEXT:  and a2, a0, a2
; RV64I-NEXT:  slliw a2, a2, 8
; RV64I-NEXT:  slliw a0, a0, 24
; RV64I-NEXT:  or a0, a0, a2
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: bswap_i32:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT:  rev8 a0, a0
; RV64ZBB-NEXT:  srai a0, a0, 32

```

```

; RV64ZBB-NEXT:  ret
%1 = tail call i32 @llvm.bswap.i32(i32 %a)
ret i32 %1
}

; Similar to bswap_i32 but the result is not sign extended.
define void @bswap_i32_nosext(i32 signext %a, ptr %x) nounwind {
; RV64I-LABEL: bswap_i32_nosext:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srlw a2, a0, 8
; RV64I-NEXT:  lui a3, 16
; RV64I-NEXT:  addi a3, a3, -256
; RV64I-NEXT:  and a2, a2, a3
; RV64I-NEXT:  srlw a4, a0, 24
; RV64I-NEXT:  or a2, a2, a4
; RV64I-NEXT:
    and a3, a0, a3
; RV64I-NEXT:  slli a3, a3, 8
; RV64I-NEXT:  slli a0, a0, 24
; RV64I-NEXT:  or a0, a0, a3
; RV64I-NEXT:  or a0, a0, a2
; RV64I-NEXT:  sw a0, 0(a1)
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: bswap_i32_nosext:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT:  rev8 a0, a0
; RV64ZBB-NEXT:  srli a0, a0, 32
; RV64ZBB-NEXT:  sw a0, 0(a1)
; RV64ZBB-NEXT:  ret
%1 = tail call i32 @llvm.bswap.i32(i32 %a)
store i32 %1, ptr %x
ret void
}

declare i64 @llvm.bswap.i64(i64)

define i64 @bswap_i64(i64 %a) {
; RV64I-LABEL: bswap_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srli a1, a0, 40
; RV64I-NEXT:  lui a2, 16
; RV64I-NEXT:  addiw a2, a2, -256
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  srli a3, a0, 56
; RV64I-NEXT:  or a1, a1, a3
; RV64I-NEXT:  srli a3, a0, 24
; RV64I-NEXT:  lui a4, 4080

```

```

; RV64I-NEXT: and a3, a3, a4
; RV64I-NEXT: srli a5, a0, 8
; RV64I-NEXT: srliw a5, a5, 24
; RV64I-NEXT: slli a5, a5, 24
; RV64I-NEXT: or a3, a5, a3
; RV64I-NEXT:
    or a1, a3, a1
; RV64I-NEXT: and a4, a0, a4
; RV64I-NEXT: slli a4, a4, 24
; RV64I-NEXT: srliw a3, a0, 24
; RV64I-NEXT: slli a3, a3, 32
; RV64I-NEXT: or a3, a4, a3
; RV64I-NEXT: and a2, a0, a2
; RV64I-NEXT: slli a2, a2, 40
; RV64I-NEXT: slli a0, a0, 56
; RV64I-NEXT: or a0, a0, a2
; RV64I-NEXT: or a0, a0, a3
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: ret
;
; RV64ZBB-LABEL: bswap_i64:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT: rev8 a0, a0
; RV64ZBB-NEXT: ret
%1 = call i64 @llvm.bswap.i64(i64 %a)
ret i64 %1
}
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64I
; RUN: llc -mtriple=riscv64 -mattr=+zbkb -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64ZBKB

```

```

define signext i32 @pack_i32(i32 signext %a, i32 signext %b) nounwind {

```

```

; RV64I-LABEL: pack_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT: slli a0, a0, 48
; RV64I-NEXT: srli a0, a0, 48
; RV64I-NEXT: slliw a1, a1, 16
; RV64I-NEXT: or a0, a1, a0
; RV64I-NEXT: ret
;
; RV64ZBKB-LABEL: pack_i32:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT: packw a0, a0, a1
; RV64ZBKB-NEXT: ret
%shl = and i32 %a, 65535
%shl1 = shl i32 %b, 16

```

```

%or = or i32 %shl1, %shl
ret i32 %or
}

define signext i32 @pack_i32_2(i16 zeroext %a, i16 zeroext %b) nounwind {
; RV64I-LABEL: pack_i32_2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slliw
a1, a1, 16
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBKB-LABEL: pack_i32_2:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  packw a0, a0, a1
; RV64ZBKB-NEXT:  ret
%zexta = zext i16 %a to i32
%zextb = zext i16 %b to i32
%shl1 = shl i32 %zextb, 16
%or = or i32 %shl1, %zexta
ret i32 %or
}

; Test case where we don't have a sign_extend_inreg after the or.
define signext i32 @pack_i32_3(i16 zeroext %0, i16 zeroext %1, i32 signext %2) {
; RV64I-LABEL: pack_i32_3:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 16
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  addw a0, a0, a2
; RV64I-NEXT:  ret
;
; RV64ZBKB-LABEL: pack_i32_3:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  packw a0, a1, a0
; RV64ZBKB-NEXT:  addw a0, a0, a2
; RV64ZBKB-NEXT:  ret
%4 = zext i16 %0 to i32
%5 = shl nuw i32 %4, 16
%6 = zext i16 %1 to i32
%7 = or i32 %5, %6
%8 = add i32 %7, %2
ret i32 %8
}

define i64 @pack_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: pack_i64:
;

```

```

RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBKB-LABEL: pack_i64:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  pack a0, a0, a1
; RV64ZBKB-NEXT:  ret
%shl = and i64 %a, 4294967295
%shl1 = shl i64 %b, 32
%or = or i64 %shl1, %shl
ret i64 %or
}

define i64 @pack_i64_2(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: pack_i64_2:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBKB-LABEL: pack_i64_2:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  pack a0, a0, a1
; RV64ZBKB-NEXT:  ret
%zexta = zext i32 %a to i64
%zextb = zext i32 %b to i64
%shl1 = shl i64 %zextb, 32
%or = or i64 %shl1, %zexta
ret i64 %or
}

define i64 @pack_i64_3(ptr %0, ptr %1) {
; RV64I-LABEL: pack_i64_3:
; RV64I:    #
%bb.0:
; RV64I-NEXT:  lw a0, 0(a0)
; RV64I-NEXT:  lwu a1, 0(a1)
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBKB-LABEL: pack_i64_3:

```

```

; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  lw a0, 0(a0)
; RV64ZBKB-NEXT:  lwu a1, 0(a1)
; RV64ZBKB-NEXT:  pack a0, a1, a0
; RV64ZBKB-NEXT:  ret
%3 = load i32, ptr %0, align 4
%4 = zext i32 %3 to i64
%5 = shl i64 %4, 32
%6 = load i32, ptr %1, align 4
%7 = zext i32 %6 to i64
%8 = or i64 %5, %7
ret i64 %8
}

```

```

define signext i32 @packh_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: packh_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT:  andi a0, a0, 255
; RV64I-NEXT:  slli a1, a1, 56
; RV64I-NEXT:  srli a1, a1, 48
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBKB-LABEL: packh_i32:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  packh a0, a0, a1
; RV64ZBKB-NEXT:  ret
%and = and i32 %a, 255
%and1 = shl i32 %b, 8
%shl = and i32 %and1, 65280
%or = or
i32 %shl, %and
ret i32 %or
}

```

```

define i32 @packh_i32_2(i32 %a, i32 %b) nounwind {
; RV64I-LABEL: packh_i32_2:
; RV64I:    # %bb.0:
; RV64I-NEXT:  andi a0, a0, 255
; RV64I-NEXT:  andi a1, a1, 255
; RV64I-NEXT:  slliw a1, a1, 8
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBKB-LABEL: packh_i32_2:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  packh a0, a0, a1
; RV64ZBKB-NEXT:  ret

```



```

%and = and i32 %a, 255
%and1 = and i32 %b, 255
%shl = shl i32 %and1, 8
%or = or i32 %shl, %and
ret i32 %or
}

```

```

define i64 @packh_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: packh_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT: andi a0, a0, 255
; RV64I-NEXT: slli a1, a1, 56
; RV64I-NEXT: srli a1, a1, 48
; RV64I-NEXT: or a0, a1, a0
; RV64I-NEXT: ret
;
; RV64ZBKB-LABEL: packh_i64:
; RV64ZBKB:   # %bb.0:
; RV64ZBKB-NEXT: packh a0, a0, a1
; RV64ZBKB-NEXT: ret
%and = and i64 %a, 255
%and1 = shl i64 %b, 8
%shl = and i64 %and1, 65280
%or = or i64 %shl,
%and
ret i64 %or
}

```

```

define i64 @packh_i64_2(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: packh_i64_2:
; RV64I:      # %bb.0:
; RV64I-NEXT: andi a0, a0, 255
; RV64I-NEXT: andi a1, a1, 255
; RV64I-NEXT: slli a1, a1, 8
; RV64I-NEXT: or a0, a1, a0
; RV64I-NEXT: ret
;
; RV64ZBKB-LABEL: packh_i64_2:
; RV64ZBKB:   # %bb.0:
; RV64ZBKB-NEXT: packh a0, a0, a1
; RV64ZBKB-NEXT: ret
%and = and i64 %a, 255
%and1 = and i64 %b, 255
%shl = shl i64 %and1, 8
%or = or i64 %shl, %and
ret i64 %or
}

```

```

define zeroext i16 @packh_i16(i8 zeroext %a, i8 zeroext %b) nounwind {
; RV64I-LABEL: packh_i16:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slliw a1, a1, 8
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBKB-LABEL: packh_i16:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  packh a0, a0, a1
; RV64ZBKB-NEXT:  ret
%zext = zext i8 %a to i16
%zext1 = zext i8 %b to i16
%shl = shl i16 %zext1, 8
%or = or i16 %shl, %zext
ret i16 %or
}

```

```

define zeroext i16 @packh_i16_2(i8
zeroext %0, i8 zeroext %1, i8 zeroext %2) {
; RV64I-LABEL: packh_i16_2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  slli a0, a0, 8
; RV64I-NEXT:  or a0, a0, a2
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srli a0, a0, 48
; RV64I-NEXT:  ret
;
; RV64ZBKB-LABEL: packh_i16_2:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  add a0, a1, a0
; RV64ZBKB-NEXT:  packh a0, a2, a0
; RV64ZBKB-NEXT:  ret
%4 = add i8 %1, %0
%5 = zext i8 %4 to i16
%6 = shl i16 %5, 8
%7 = zext i8 %2 to i16
%8 = or i16 %6, %7
ret i16 %8
}

```

```

define i64 @pack_i64_allWUsers(i32 signext %0, i32 signext %1, i32 signext %2) {
; RV64I-LABEL: pack_i64_allWUsers:
; RV64I:      # %bb.0:
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  slli a2, a2, 32

```

```

; RV64I-NEXT: srli a2, a2, 32
; RV64I-NEXT: or a0, a0, a2
; RV64I-NEXT: ret
;
; RV64ZBKB-LABEL: pack_i64_allWUsers:
; RV64ZBKB: # %bb.0:
; RV64ZBKB-NEXT: add a0, a1, a0
; RV64ZBKB-NEXT:
    pack a0, a2, a0
; RV64ZBKB-NEXT: ret
%4 = add i32 %1, %0
%5 = zext i32 %4 to i64
%6 = shl i64 %5, 32
%7 = zext i32 %2 to i64
%8 = or i64 %6, %7
ret i64 %8
}

define signext i32 @pack_i32_allWUsers(i16 zeroext %0, i16 zeroext %1, i16 zeroext %2) {
; RV64I-LABEL: pack_i32_allWUsers:
; RV64I: # %bb.0:
; RV64I-NEXT: add a0, a1, a0
; RV64I-NEXT: slliw a0, a0, 16
; RV64I-NEXT: or a0, a0, a2
; RV64I-NEXT: ret
;
; RV64ZBKB-LABEL: pack_i32_allWUsers:
; RV64ZBKB: # %bb.0:
; RV64ZBKB-NEXT: add a0, a1, a0
; RV64ZBKB-NEXT: packw a0, a2, a0
; RV64ZBKB-NEXT: ret
%4 = add i16 %1, %0
%5 = zext i16 %4 to i32
%6 = shl i32 %5, 16
%7 = zext i16 %2 to i32
%8 = or i32 %6, %7
ret i32 %8
}

define i64 @pack_i64_imm() {
; RV64I-LABEL: pack_i64_imm:
; RV64I: # %bb.0:
; RV64I-NEXT: lui a0, 65793
; RV64I-NEXT: addiw a0, a0, 16
; RV64I-NEXT: slli a1, a0, 32
; RV64I-NEXT: add a0, a0, a1
; RV64I-NEXT: ret
;

```

```

; RV64ZBKB-LABEL:
pack_i64_imm:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  lui a0, 65793
; RV64ZBKB-NEXT:  addi a0, a0, 16
; RV64ZBKB-NEXT:  pack a0, a0, a0
; RV64ZBKB-NEXT:  ret
ret i64 1157442765409226768 ; 0x0101010101010101
}
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -verify-machineinstrs < %s \
; RUN:  -riscv-experimental-rv64-legal-i32 | FileCheck -check-prefix=RV64I %s

```

; Check indexed and unindexed, sext, zext and anyext loads

```

define void @lb(ptr %a, ptr %b) nounwind {
; RV64I-LABEL: lb:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lb a2, 1(a0)
; RV64I-NEXT:  lbu zero, 0(a0)
; RV64I-NEXT:  sw a2, 0(a1)
; RV64I-NEXT:  ret
%1 = getelementptr i8, ptr %a, i32 1
%2 = load i8, ptr %1
%3 = sext i8 %2 to i32
; the unused load will produce an anyext for selection
%4 = load volatile i8, ptr %a
store i32 %3, ptr %b
ret void
}

```

```

define void @lbu(ptr %a, ptr %b) nounwind {
; RV64I-LABEL: lbu:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lbu a0, 1(a0)
; RV64I-NEXT:  sw a0, 0(a1)
; RV64I-NEXT:  ret
%1 = getelementptr i8, ptr %a, i32 1
%2 = load i8, ptr %1
%3 = zext i8 %2 to i32
store i32 %3, ptr %b
ret void
}

```

```

define void @lh(ptr
%a, ptr %b) nounwind {
; RV64I-LABEL: lh:
; RV64I:    # %bb.0:

```

```

; RV64I-NEXT:  lh a2, 2(a0)
; RV64I-NEXT:  lh zero, 0(a0)
; RV64I-NEXT:  sw a2, 0(a1)
; RV64I-NEXT:  ret
%1 = getelementptr i16, ptr %a, i32 1
%2 = load i16, ptr %1
%3 = sext i16 %2 to i32
; the unused load will produce an anyext for selection
%4 = load volatile i16, ptr %a
store i32 %3, ptr %b
ret void
}

```

```

define void @lhu(ptr %a, ptr %b) nounwind {
; RV64I-LABEL: lhu:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lhu a0, 2(a0)
; RV64I-NEXT:  sw a0, 0(a1)
; RV64I-NEXT:  ret
%1 = getelementptr i16, ptr %a, i32 1
%2 = load i16, ptr %1
%3 = zext i16 %2 to i32
store i32 %3, ptr %b
ret void
}

```

```

define void @lw(ptr %a, ptr %b) nounwind {
; RV64I-LABEL: lw:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lw a2, 4(a0)
; RV64I-NEXT:  lw zero, 0(a0)
; RV64I-NEXT:  sd a2, 0(a1)
; RV64I-NEXT:  ret
%1 = getelementptr i32, ptr %a, i64 1
%2 = load i32, ptr %1
%3 = sext i32 %2 to i64
; the unused load
will produce an anyext for selection
%4 = load volatile i32, ptr %a
store i64 %3, ptr %b
ret void
}

```

```

define void @lwu(ptr %a, ptr %b) nounwind {
; RV64I-LABEL: lwu:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lwu a0, 4(a0)
; RV64I-NEXT:  sd a0, 0(a1)

```

```

; RV64I-NEXT:  ret
%1 = getelementptr i32, ptr %a, i64 1
%2 = load i32, ptr %1
%3 = zext i32 %2 to i64
store i64 %3, ptr %b
ret void
}
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -riscv-disable-using-constant-pool-for-large-ints -verify-machineinstrs < %s \
; RUN:  -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefixes=RV64I,RV64I-NOPOOL
; RUN: llc -mtriple=riscv64 -verify-machineinstrs < %s \
; RUN:  -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefixes=RV64I,RV64I-POOL
; RUN: llc -mtriple=riscv64 -riscv-disable-using-constant-pool-for-large-ints -mattr=+zba \
; RUN:  -riscv-experimental-rv64-legal-i32 -verify-machineinstrs < %s | FileCheck %s -check-prefix=RV64IZBA
; RUN: llc -mtriple=riscv64 -riscv-disable-using-constant-pool-for-large-ints -mattr=+zbb \
; RUN:  -riscv-experimental-rv64-legal-i32 -verify-machineinstrs < %s | FileCheck %s -check-prefix=RV64IZBB
; RUN: llc -mtriple=riscv64 -riscv-disable-using-constant-pool-for-large-ints -mattr=+zbs \
; RUN:  -riscv-experimental-rv64-legal-i32 -verify-machineinstrs < %s | FileCheck
% s -check-prefix=RV64IZBS
; RUN: llc -mtriple=riscv64 -riscv-disable-using-constant-pool-for-large-ints -mattr=+xtheadbb \
; RUN:  -riscv-experimental-rv64-legal-i32 -verify-machineinstrs < %s | FileCheck %s -check-
prefix=RV64IXTHEADBB

```

; Materializing constants

; TODO: It would be preferable if anyext constant returns were sign rather  
; than zero extended. See PR39092. For now, mark returns as explicitly signext  
; (this matches what Clang would generate for equivalent C/C++ anyway).

```

define signext i32 @zero() nounwind {
; RV64I-LABEL: zero:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a0, 0
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: zero:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  li a0, 0
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: zero:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  li a0, 0
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: zero:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  li a0, 0

```

```

; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: zero:
; RV64IXTHEADBB:   # %bb.0:
;
RV64IXTHEADBB-NEXT:  li a0, 0
; RV64IXTHEADBB-NEXT:  ret
ret i32 0
}

```

```

define signext i32 @pos_small() nounwind {
; RV64I-LABEL: pos_small:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a0, 2047
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: pos_small:
; RV64IZBA:   # %bb.0:
; RV64IZBA-NEXT:  li a0, 2047
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: pos_small:
; RV64IZBB:   # %bb.0:
; RV64IZBB-NEXT:  li a0, 2047
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: pos_small:
; RV64IZBS:   # %bb.0:
; RV64IZBS-NEXT:  li a0, 2047
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: pos_small:
; RV64IXTHEADBB:   # %bb.0:
; RV64IXTHEADBB-NEXT:  li a0, 2047
; RV64IXTHEADBB-NEXT:  ret
ret i32 2047
}

```

```

define signext i32 @neg_small() nounwind {
; RV64I-LABEL: neg_small:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a0, -2048
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: neg_small:
; RV64IZBA:   # %bb.0:
; RV64IZBA-NEXT:  li a0, -2048
; RV64IZBA-NEXT:  ret

```

```

;
; RV64IZBB-LABEL:
neg_small:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  li a0, -2048
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: neg_small:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  li a0, -2048
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: neg_small:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  li a0, -2048
; RV64IXTHEADBB-NEXT:  ret
ret i32 -2048
}

define signext i32 @pos_i32() nounwind {
; RV64I-LABEL: pos_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 423811
; RV64I-NEXT:  addiw a0, a0, -1297
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: pos_i32:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 423811
; RV64IZBA-NEXT:  addiw a0, a0, -1297
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: pos_i32:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 423811
; RV64IZBB-NEXT:  addiw a0, a0, -1297
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: pos_i32:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 423811
; RV64IZBS-NEXT:  addiw a0, a0, -1297
; RV64IZBS-NEXT:
ret
;
; RV64IXTHEADBB-LABEL: pos_i32:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 423811

```



```
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -1297
; RV64IXTHEADBB-NEXT:  ret
ret i32 1735928559
}
```

```
define signext i32 @neg_i32() nounwind {
; RV64I-LABEL: neg_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a0, 912092
; RV64I-NEXT:  addiw a0, a0, -273
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: neg_i32:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 912092
; RV64IZBA-NEXT:  addiw a0, a0, -273
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: neg_i32:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 912092
; RV64IZBB-NEXT:  addiw a0, a0, -273
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: neg_i32:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 912092
; RV64IZBS-NEXT:  addiw a0, a0, -273
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: neg_i32:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 912092
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -273
;
RV64IXTHEADBB-NEXT:  ret
ret i32 -559038737
}
```

```
define signext i32 @pos_i32_hi20_only() nounwind {
; RV64I-LABEL: pos_i32_hi20_only:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a0, 16
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: pos_i32_hi20_only:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 16
```

```

; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: pos_i32_hi20_only:
; RV64IZBB:      # %bb.0:
; RV64IZBB-NEXT:  lui a0, 16
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: pos_i32_hi20_only:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT:  lui a0, 16
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: pos_i32_hi20_only:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 16
; RV64IXTHEADBB-NEXT:  ret
ret i32 65536 ; 0x10000
}

define signext i32 @neg_i32_hi20_only() nounwind {
; RV64I-LABEL: neg_i32_hi20_only:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a0, 1048560
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: neg_i32_hi20_only:
; RV64IZBA:      # %bb.0:
; RV64IZBA-NEXT:  lui a0,
1048560
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: neg_i32_hi20_only:
; RV64IZBB:      # %bb.0:
; RV64IZBB-NEXT:  lui a0, 1048560
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: neg_i32_hi20_only:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT:  lui a0, 1048560
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: neg_i32_hi20_only:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 1048560
; RV64IXTHEADBB-NEXT:  ret
ret i32 -65536 ; -0x10000
}

```

; This can be materialized with ADDI+SLLI, improving compressibility.

```
define signext i32 @imm_left_shifted_addi() nounwind {
; RV64I-LABEL: imm_left_shifted_addi:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a0, 32
; RV64I-NEXT:  addiw a0, a0, -64
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_left_shifted_addi:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 32
; RV64IZBA-NEXT:  addiw a0, a0, -64
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_left_shifted_addi:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 32
;
; RV64IZBB-NEXT:  addiw a0, a0, -64
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_left_shifted_addi:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 32
; RV64IZBS-NEXT:  addiw a0, a0, -64
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_left_shifted_addi:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 32
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -64
; RV64IXTHEADBB-NEXT:  ret
ret i32 131008 ; 0x1FFC0
}
```

; This can be materialized with ADDI+SRLI, improving compressibility.

```
define signext i32 @imm_right_shifted_addi() nounwind {
; RV64I-LABEL: imm_right_shifted_addi:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a0, 524288
; RV64I-NEXT:  addiw a0, a0, -1
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_right_shifted_addi:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 524288
```

```

; RV64IZBA-NEXT:  addiw a0, a0, -1
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_right_shifted_addi:
; RV64IZBB:      # %bb.0:
; RV64IZBB-NEXT:  lui a0, 524288
; RV64IZBB-NEXT:
    addiw a0, a0, -1
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_right_shifted_addi:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT:  lui a0, 524288
; RV64IZBS-NEXT:  addiw a0, a0, -1
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_right_shifted_addi:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 524288
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -1
; RV64IXTHEADBB-NEXT:  ret
ret i32 2147483647 ; 0x7FFFFFFF
}

```

; This can be materialized with LUI+SRLI, improving compressibility.

```

define signext i32 @imm_right_shifted_lui() nounwind {
; RV64I-LABEL: imm_right_shifted_lui:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a0, 56
; RV64I-NEXT:  addiw a0, a0, 580
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_right_shifted_lui:
; RV64IZBA:      # %bb.0:
; RV64IZBA-NEXT:  lui a0, 56
; RV64IZBA-NEXT:  addiw a0, a0, 580
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_right_shifted_lui:
; RV64IZBB:      # %bb.0:
; RV64IZBB-NEXT:  lui a0, 56
; RV64IZBB-NEXT:
    addiw a0, a0, 580
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_right_shifted_lui:
; RV64IZBS:      # %bb.0:

```

```

; RV64IZBS-NEXT:  lui a0, 56
; RV64IZBS-NEXT:  addiw a0, a0, 580
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_right_shifted_lui:
; RV64IXTHEADBB:   # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 56
; RV64IXTHEADBB-NEXT:  addiw a0, a0, 580
; RV64IXTHEADBB-NEXT:  ret
ret i32 229956 ; 0x38244
}

```

```

define i64 @imm64_1() nounwind {
; RV64I-LABEL: imm64_1:
; RV64I:       # %bb.0:
; RV64I-NEXT:  li a0, 1
; RV64I-NEXT:  slli a0, a0, 31
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm64_1:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  li a0, 1
; RV64IZBA-NEXT:  slli a0, a0, 31
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm64_1:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  li a0, 1
; RV64IZBB-NEXT:  slli a0, a0, 31
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm64_1:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  bseti a0, zero, 31
; RV64IZBS-NEXT:  ret
;
;
RV64IXTHEADBB-LABEL: imm64_1:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  li a0, 1
; RV64IXTHEADBB-NEXT:  slli a0, a0, 31
; RV64IXTHEADBB-NEXT:  ret
ret i64 2147483648 ; 0x8000_0000
}

```

```

define i64 @imm64_2() nounwind {
; RV64I-LABEL: imm64_2:
; RV64I:       # %bb.0:

```

```

; RV64I-NEXT:  li a0, -1
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm64_2:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  li a0, -1
; RV64IZBA-NEXT:  srli a0, a0, 32
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm64_2:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  li a0, -1
; RV64IZBB-NEXT:  srli a0, a0, 32
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm64_2:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  li a0, -1
; RV64IZBS-NEXT:  srli a0, a0, 32
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm64_2:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  li a0, -1
; RV64IXTHEADBB-NEXT:  srli a0, a0, 32
; RV64IXTHEADBB-NEXT:  ret
ret i64 4294967295 ;
0xFFFF_FFFF
}

```

```

define i64 @imm64_3() nounwind {
; RV64I-LABEL: imm64_3:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a0, 1
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm64_3:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  li a0, 1
; RV64IZBA-NEXT:  slli a0, a0, 32
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm64_3:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  li a0, 1
; RV64IZBB-NEXT:  slli a0, a0, 32

```

```

; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm64_3:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT:  bseti a0, zero, 32
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm64_3:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  li a0, 1
; RV64IXTHEADBB-NEXT:  slli a0, a0, 32
; RV64IXTHEADBB-NEXT:  ret
ret i64 4294967296 ; 0x1_0000_0000
}

define i64 @imm64_4() nounwind {
; RV64I-LABEL: imm64_4:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a0, -1
; RV64I-NEXT:  slli a0, a0, 63
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm64_4:
; RV64IZBA:
;   # %bb.0:
; RV64IZBA-NEXT:  li a0, -1
; RV64IZBA-NEXT:  slli a0, a0, 63
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm64_4:
; RV64IZBB:      # %bb.0:
; RV64IZBB-NEXT:  li a0, -1
; RV64IZBB-NEXT:  slli a0, a0, 63
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm64_4:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT:  bseti a0, zero, 63
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm64_4:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  li a0, -1
; RV64IXTHEADBB-NEXT:  slli a0, a0, 63
; RV64IXTHEADBB-NEXT:  ret
ret i64 9223372036854775808 ; 0x8000_0000_0000_0000
}

```

```

define i64 @imm64_5() nounwind {
; RV64I-LABEL: imm64_5:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a0, -1
; RV64I-NEXT:  slli a0, a0, 63
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm64_5:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  li a0, -1
; RV64IZBA-NEXT:  slli a0, a0, 63
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm64_5:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  li a0, -1
; RV64IZBB-NEXT:
;               slli a0, a0, 63
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm64_5:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  bseti a0, zero, 63
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm64_5:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  li a0, -1
; RV64IXTHEADBB-NEXT:  slli a0, a0, 63
; RV64IXTHEADBB-NEXT:  ret
ret i64 -9223372036854775808 ; 0x8000_0000_0000_0000
}

```

```

define i64 @imm64_6() nounwind {
; RV64I-LABEL: imm64_6:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a0, 9321
; RV64I-NEXT:  addi a0, a0, -1329
; RV64I-NEXT:  slli a0, a0, 35
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm64_6:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 9321
; RV64IZBA-NEXT:  addi a0, a0, -1329
; RV64IZBA-NEXT:  slli a0, a0, 35
; RV64IZBA-NEXT:  ret
;

```



```

; RV64IZBB-LABEL: imm64_6:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 9321
; RV64IZBB-NEXT:  addi a0, a0, -1329
; RV64IZBB-NEXT:  slli a0, a0, 35
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm64_6:
;
RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 9321
; RV64IZBS-NEXT:  addi a0, a0, -1329
; RV64IZBS-NEXT:  slli a0, a0, 35
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm64_6:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 9321
; RV64IXTHEADBB-NEXT:  addi a0, a0, -1329
; RV64IXTHEADBB-NEXT:  slli a0, a0, 35
; RV64IXTHEADBB-NEXT:  ret
ret i64 1311768464867721216 ; 0x1234_5678_0000_0000
}

```

```

define i64 @imm64_7() nounwind {
; RV64I-LABEL: imm64_7:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a0, 7
; RV64I-NEXT:  slli a0, a0, 36
; RV64I-NEXT:  addi a0, a0, 11
; RV64I-NEXT:  slli a0, a0, 24
; RV64I-NEXT:  addi a0, a0, 15
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm64_7:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  li a0, 7
; RV64IZBA-NEXT:  slli a0, a0, 36
; RV64IZBA-NEXT:  addi a0, a0, 11
; RV64IZBA-NEXT:  slli a0, a0, 24
; RV64IZBA-NEXT:  addi a0, a0, 15
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm64_7:
; RV64IZBB:    #
%bb.0:
; RV64IZBB-NEXT:  li a0, 7
; RV64IZBB-NEXT:  slli a0, a0, 36

```

```

; RV64IZBB-NEXT:  addi a0, a0, 11
; RV64IZBB-NEXT:  slli a0, a0, 24
; RV64IZBB-NEXT:  addi a0, a0, 15
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm64_7:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT:  li a0, 7
; RV64IZBS-NEXT:  slli a0, a0, 36
; RV64IZBS-NEXT:  addi a0, a0, 11
; RV64IZBS-NEXT:  slli a0, a0, 24
; RV64IZBS-NEXT:  addi a0, a0, 15
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm64_7:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  li a0, 7
; RV64IXTHEADBB-NEXT:  slli a0, a0, 36
; RV64IXTHEADBB-NEXT:  addi a0, a0, 11
; RV64IXTHEADBB-NEXT:  slli a0, a0, 24
; RV64IXTHEADBB-NEXT:  addi a0, a0, 15
; RV64IXTHEADBB-NEXT:  ret
ret i64 8070450532432478223 ; 0x7000_0000_0B00_000F
}

```

; TODO: it can be preferable to put constants that are expensive to materialise  
into the constant pool, especially for -Os.

```

define i64 @imm64_8() nounwind {
; RV64-NOPOOL-LABEL: imm64_8:
; RV64-NOPOOL:
    # %bb.0:
; RV64-NOPOOL-NEXT:  lui a0, 583
; RV64-NOPOOL-NEXT:  addiw a0, a0, -1875
; RV64-NOPOOL-NEXT:  slli a0, a0, 14
; RV64-NOPOOL-NEXT:  addi a0, a0, -947
; RV64-NOPOOL-NEXT:  slli a0, a0, 12
; RV64-NOPOOL-NEXT:  addi a0, a0, 1511
; RV64-NOPOOL-NEXT:  slli a0, a0, 13
; RV64-NOPOOL-NEXT:  addi a0, a0, -272
; RV64-NOPOOL-NEXT:  ret
;
; RV64I-POOL-LABEL: imm64_8:
; RV64I-POOL:      # %bb.0:
; RV64I-POOL-NEXT:  lui a0, %hi(.LCPI17_0)
; RV64I-POOL-NEXT:  ld a0, %lo(.LCPI17_0)(a0)
; RV64I-POOL-NEXT:  ret
;
; RV64IZBA-LABEL: imm64_8:

```

```

; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT: lui a0, 596523
; RV64IZBA-NEXT: addi a0, a0, 965
; RV64IZBA-NEXT: slli.uw a0, a0, 13
; RV64IZBA-NEXT: addi a0, a0, -1347
; RV64IZBA-NEXT: slli a0, a0, 12
; RV64IZBA-NEXT: addi a0, a0, -529
; RV64IZBA-NEXT: slli a0, a0, 4
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: imm64_8:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT: lui a0, 583
; RV64IZBB-NEXT: addiw
a0, a0, -1875
; RV64IZBB-NEXT: slli a0, a0, 14
; RV64IZBB-NEXT: addi a0, a0, -947
; RV64IZBB-NEXT: slli a0, a0, 12
; RV64IZBB-NEXT: addi a0, a0, 1511
; RV64IZBB-NEXT: slli a0, a0, 13
; RV64IZBB-NEXT: addi a0, a0, -272
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: imm64_8:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT: lui a0, 583
; RV64IZBS-NEXT: addiw a0, a0, -1875
; RV64IZBS-NEXT: slli a0, a0, 14
; RV64IZBS-NEXT: addi a0, a0, -947
; RV64IZBS-NEXT: slli a0, a0, 12
; RV64IZBS-NEXT: addi a0, a0, 1511
; RV64IZBS-NEXT: slli a0, a0, 13
; RV64IZBS-NEXT: addi a0, a0, -272
; RV64IZBS-NEXT: ret
;
; RV64IXTHEADBB-LABEL: imm64_8:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT: lui a0, 583
; RV64IXTHEADBB-NEXT: addiw a0, a0, -1875
; RV64IXTHEADBB-NEXT: slli a0, a0, 14
; RV64IXTHEADBB-NEXT: addi a0, a0, -947
; RV64IXTHEADBB-NEXT: slli a0, a0, 12
; RV64IXTHEADBB-NEXT: addi a0, a0, 1511
; RV64IXTHEADBB-NEXT: slli a0, a0, 13
; RV64IXTHEADBB-NEXT:
addi a0, a0, -272
; RV64IXTHEADBB-NEXT: ret
ret i64 1311768467463790320 ; 0x1234_5678_9ABC_DEF0

```

```
}
```

```
define i64 @imm64_9() nounwind {  
; RV64I-LABEL: imm64_9:  
; RV64I:    # %bb.0:  
; RV64I-NEXT:  li a0, -1  
; RV64I-NEXT:  ret  
;  
; RV64IZBA-LABEL: imm64_9:  
; RV64IZBA:    # %bb.0:  
; RV64IZBA-NEXT:  li a0, -1  
; RV64IZBA-NEXT:  ret  
;  
; RV64IZBB-LABEL: imm64_9:  
; RV64IZBB:    # %bb.0:  
; RV64IZBB-NEXT:  li a0, -1  
; RV64IZBB-NEXT:  ret  
;  
; RV64IZBS-LABEL: imm64_9:  
; RV64IZBS:    # %bb.0:  
; RV64IZBS-NEXT:  li a0, -1  
; RV64IZBS-NEXT:  ret  
;  
; RV64IXTHEADBB-LABEL: imm64_9:  
; RV64IXTHEADBB:    # %bb.0:  
; RV64IXTHEADBB-NEXT:  li a0, -1  
; RV64IXTHEADBB-NEXT:  ret  
ret i64 -1  
}
```

; Various cases where extraneous ADDIs can be inserted where a (left shifted)  
; LUI suffices.

```
define i64 @imm_left_shifted_lui_1() nounwind {  
; RV64I-LABEL: imm_left_shifted_lui_1:  
; RV64I:    # %bb.0:  
; RV64I-NEXT:  lui a0, 262145  
; RV64I-NEXT:  slli a0, a0, 1  
;  
RV64I-NEXT:  ret  
;  
; RV64IZBA-LABEL: imm_left_shifted_lui_1:  
; RV64IZBA:    # %bb.0:  
; RV64IZBA-NEXT:  lui a0, 262145  
; RV64IZBA-NEXT:  slli a0, a0, 1  
; RV64IZBA-NEXT:  ret  
;  
; RV64IZBB-LABEL: imm_left_shifted_lui_1:
```

```

; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 262145
; RV64IZBB-NEXT:  slli a0, a0, 1
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_left_shifted_lui_1:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 262145
; RV64IZBS-NEXT:  slli a0, a0, 1
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_left_shifted_lui_1:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 262145
; RV64IXTHEADBB-NEXT:  slli a0, a0, 1
; RV64IXTHEADBB-NEXT:  ret
ret i64 2147491840 ; 0x8000_2000
}

```

```

define i64 @imm_left_shifted_lui_2() nounwind {
; RV64I-LABEL: imm_left_shifted_lui_2:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 262145
; RV64I-NEXT:  slli a0, a0, 2
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_left_shifted_lui_2:
; RV64IZBA:
# %bb.0:
; RV64IZBA-NEXT:  lui a0, 262145
; RV64IZBA-NEXT:  slli a0, a0, 2
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_left_shifted_lui_2:
; RV64IZBB:  # %bb.0:
; RV64IZBB-NEXT:  lui a0, 262145
; RV64IZBB-NEXT:  slli a0, a0, 2
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_left_shifted_lui_2:
; RV64IZBS:  # %bb.0:
; RV64IZBS-NEXT:  lui a0, 262145
; RV64IZBS-NEXT:  slli a0, a0, 2
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_left_shifted_lui_2:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 262145

```

```
; RV64IXTHEADBB-NEXT:  slli a0, a0, 2
; RV64IXTHEADBB-NEXT:  ret
ret i64 4294983680 ; 0x1_0000_4000
}
```

```
define i64 @imm_left_shifted_lui_3() nounwind {
; RV64I-LABEL: imm_left_shifted_lui_3:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a0, 4097
; RV64I-NEXT:  slli a0, a0, 20
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_left_shifted_lui_3:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 4097
; RV64IZBA-NEXT:  slli a0, a0, 20
;
RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_left_shifted_lui_3:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 4097
; RV64IZBB-NEXT:  slli a0, a0, 20
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_left_shifted_lui_3:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 4097
; RV64IZBS-NEXT:  slli a0, a0, 20
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_left_shifted_lui_3:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 4097
; RV64IXTHEADBB-NEXT:  slli a0, a0, 20
; RV64IXTHEADBB-NEXT:  ret
ret i64 17596481011712 ; 0x1001_0000_0000
}
```

; Various cases where extraneous ADDIs can be inserted where a (right shifted)  
; LUI suffices, or where multiple ADDIs can be used instead of a single LUI.

```
define i64 @imm_right_shifted_lui_1() nounwind {
; RV64I-LABEL: imm_right_shifted_lui_1:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a0, 983056
; RV64I-NEXT:  srli a0, a0, 16
; RV64I-NEXT:  ret
```

```

;
; RV64IZBA-LABEL: imm_right_shifted_lui_1:
; RV64IZBA:
    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 983056
; RV64IZBA-NEXT:  srli a0, a0, 16
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_right_shifted_lui_1:
; RV64IZBB:      # %bb.0:
; RV64IZBB-NEXT:  lui a0, 983056
; RV64IZBB-NEXT:  srli a0, a0, 16
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_right_shifted_lui_1:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT:  lui a0, 983056
; RV64IZBS-NEXT:  srli a0, a0, 16
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_right_shifted_lui_1:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 983056
; RV64IXTHEADBB-NEXT:  srli a0, a0, 16
; RV64IXTHEADBB-NEXT:  ret
ret i64 281474976706561 ; 0xFFFF_FFFF_F001
}

```

```

define i64 @imm_right_shifted_lui_2() nounwind {
; RV64I-LABEL: imm_right_shifted_lui_2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a0, 1044481
; RV64I-NEXT:  slli a0, a0, 12
; RV64I-NEXT:  srli a0, a0, 24
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_right_shifted_lui_2:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:
    lui a0, 1044481
; RV64IZBA-NEXT:  slli a0, a0, 12
; RV64IZBA-NEXT:  srli a0, a0, 24
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_right_shifted_lui_2:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 1044481
; RV64IZBB-NEXT:  slli a0, a0, 12

```

```

; RV64IZBB-NEXT: srli a0, a0, 24
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: imm_right_shifted_lui_2:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT: lui a0, 1044481
; RV64IZBS-NEXT: slli a0, a0, 12
; RV64IZBS-NEXT: srli a0, a0, 24
; RV64IZBS-NEXT: ret
;
; RV64IXTHEADBB-LABEL: imm_right_shifted_lui_2:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT: lui a0, 1044481
; RV64IXTHEADBB-NEXT: slli a0, a0, 12
; RV64IXTHEADBB-NEXT: srli a0, a0, 24
; RV64IXTHEADBB-NEXT: ret
ret i64 1099511623681 ; 0xFF_FFFF_F001
}

```

; We can materialize the upper bits with a single (shifted) LUI, but that option  
; can be missed due to the lower bits, which aren't just 1s or just 0s.

```

define i64 @imm_decoupled_lui_addi() nounwind
{
; RV64I-LABEL: imm_decoupled_lui_addi:
; RV64I:      # %bb.0:
; RV64I-NEXT: lui a0, 4097
; RV64I-NEXT: slli a0, a0, 20
; RV64I-NEXT: addi a0, a0, -3
; RV64I-NEXT: ret
;
; RV64IZBA-LABEL: imm_decoupled_lui_addi:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT: lui a0, 4097
; RV64IZBA-NEXT: slli a0, a0, 20
; RV64IZBA-NEXT: addi a0, a0, -3
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: imm_decoupled_lui_addi:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT: lui a0, 4097
; RV64IZBB-NEXT: slli a0, a0, 20
; RV64IZBB-NEXT: addi a0, a0, -3
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: imm_decoupled_lui_addi:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT: lui a0, 4097

```



```

; RV64IZBS-NEXT:  slli a0, a0, 20
; RV64IZBS-NEXT:  addi a0, a0, -3
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_decoupled_lui_addi:
; RV64IXTHEADBB:   # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 4097
; RV64IXTHEADBB-NEXT:  slli a0, a0, 20
; RV64IXTHEADBB-NEXT:  addi a0, a0, -3
; RV64IXTHEADBB-NEXT:
    ret
ret i64 17596481011709 ; 0x1000_FFFF_FFFD
}

```

; This constant can be materialized for RV64 with LUI+SRLI+XORI.

```

define i64 @imm_end_xori_1() nounwind {
; RV64I-LABEL: imm_end_xori_1:
; RV64I:   # %bb.0:
; RV64I-NEXT:  lui a0, 983040
; RV64I-NEXT:  srli a0, a0, 3
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_end_xori_1:
; RV64IZBA:   # %bb.0:
; RV64IZBA-NEXT:  lui a0, 983040
; RV64IZBA-NEXT:  srli a0, a0, 3
; RV64IZBA-NEXT:  not a0, a0
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_end_xori_1:
; RV64IZBB:   # %bb.0:
; RV64IZBB-NEXT:  lui a0, 983040
; RV64IZBB-NEXT:  srli a0, a0, 3
; RV64IZBB-NEXT:  not a0, a0
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_end_xori_1:
; RV64IZBS:   # %bb.0:
; RV64IZBS-NEXT:  lui a0, 983040
; RV64IZBS-NEXT:  srli a0, a0, 3
; RV64IZBS-NEXT:  not a0, a0
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_end_xori_1:
; RV64IXTHEADBB:   # %bb.0:
; RV64IXTHEADBB-NEXT:

```

```

    lui a0, 983040
; RV64IXTHEADBB-NEXT: srli a0, a0, 3
; RV64IXTHEADBB-NEXT: not a0, a0
; RV64IXTHEADBB-NEXT: ret
ret i64 -2305843009180139521 ; 0xE000_0000_01FF_FFFF
}

```

; This constant can be materialized for RV64 with ADDI+SLLI+ADDI+ADDI.

```

define i64 @imm_end_2addi_1() nounwind {
; RV64I-LABEL: imm_end_2addi_1:
; RV64I:    # %bb.0:
; RV64I-NEXT: li a0, -2047
; RV64I-NEXT: slli a0, a0, 39
; RV64I-NEXT: addi a0, a0, -2048
; RV64I-NEXT: addi a0, a0, -1
; RV64I-NEXT: ret
;
; RV64IZBA-LABEL: imm_end_2addi_1:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT: li a0, -2047
; RV64IZBA-NEXT: slli a0, a0, 39
; RV64IZBA-NEXT: addi a0, a0, -2048
; RV64IZBA-NEXT: addi a0, a0, -1
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: imm_end_2addi_1:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT: li a0, -2047
; RV64IZBB-NEXT: slli a0, a0, 39
; RV64IZBB-NEXT: addi a0, a0, -2048
; RV64IZBB-NEXT: addi a0, a0, -1
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: imm_end_2addi_1:
;
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT: li a0, -2047
; RV64IZBS-NEXT: slli a0, a0, 39
; RV64IZBS-NEXT: addi a0, a0, -2048
; RV64IZBS-NEXT: addi a0, a0, -1
; RV64IZBS-NEXT: ret
;
; RV64IXTHEADBB-LABEL: imm_end_2addi_1:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT: li a0, -2047
; RV64IXTHEADBB-NEXT: slli a0, a0, 39
; RV64IXTHEADBB-NEXT: addi a0, a0, -2048

```

```
; RV64IXTHEADBB-NEXT:  addi a0, a0, -1
; RV64IXTHEADBB-NEXT:  ret
ret i64 -1125350151030785 ; 0xFFFC_007F_FFFF_F7FF
}
```

; This constant can be more efficiently materialized for RV64 if we use two registers instead of one.

```
define i64 @imm_2reg_1() nounwind {
; RV64I-LABEL: imm_2reg_1:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a0, 74565
; RV64I-NEXT:  addiw a0, a0, 1656
; RV64I-NEXT:  slli a1, a0, 57
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_2reg_1:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 74565
; RV64IZBA-NEXT:  addiw a0, a0, 1656
; RV64IZBA-NEXT:
;               slli a1, a0, 57
; RV64IZBA-NEXT:  add a0, a0, a1
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_2reg_1:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 74565
; RV64IZBB-NEXT:  addiw a0, a0, 1656
; RV64IZBB-NEXT:  slli a1, a0, 57
; RV64IZBB-NEXT:  add a0, a0, a1
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_2reg_1:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 74565
; RV64IZBS-NEXT:  addiw a0, a0, 1656
; RV64IZBS-NEXT:  slli a1, a0, 57
; RV64IZBS-NEXT:  add a0, a0, a1
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_2reg_1:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 74565
; RV64IXTHEADBB-NEXT:  addiw a0, a0, 1656
; RV64IXTHEADBB-NEXT:  slli a1, a0, 57
; RV64IXTHEADBB-NEXT:  add a0, a0, a1
```

```
; RV64IXTHEADBB-NEXT:  ret
ret i64 -1152921504301427080 ; 0xF000_0000_1234_5678
}
```

; FIXME: This should use a single ADDI for the immediate.

```
define void @imm_store_i16_neg1(ptr %p) nounwind {
```

```
; RV64I-LABEL: imm_store_i16_neg1:
```

```
;
```

```
RV64I:    # %bb.0:
```

```
; RV64I-NEXT:  li a1, -1
```

```
; RV64I-NEXT:  sh a1, 0(a0)
```

```
; RV64I-NEXT:  ret
```

```
;
```

```
; RV64IZBA-LABEL: imm_store_i16_neg1:
```

```
; RV64IZBA:    # %bb.0:
```

```
; RV64IZBA-NEXT:  li a1, -1
```

```
; RV64IZBA-NEXT:  sh a1, 0(a0)
```

```
; RV64IZBA-NEXT:  ret
```

```
;
```

```
; RV64IZBB-LABEL: imm_store_i16_neg1:
```

```
; RV64IZBB:    # %bb.0:
```

```
; RV64IZBB-NEXT:  li a1, -1
```

```
; RV64IZBB-NEXT:  sh a1, 0(a0)
```

```
; RV64IZBB-NEXT:  ret
```

```
;
```

```
; RV64IZBS-LABEL: imm_store_i16_neg1:
```

```
; RV64IZBS:    # %bb.0:
```

```
; RV64IZBS-NEXT:  li a1, -1
```

```
; RV64IZBS-NEXT:  sh a1, 0(a0)
```

```
; RV64IZBS-NEXT:  ret
```

```
;
```

```
; RV64IXTHEADBB-LABEL: imm_store_i16_neg1:
```

```
; RV64IXTHEADBB:    # %bb.0:
```

```
; RV64IXTHEADBB-NEXT:  li a1, -1
```

```
; RV64IXTHEADBB-NEXT:  sh a1, 0(a0)
```

```
; RV64IXTHEADBB-NEXT:  ret
```

```
store i16 -1, ptr %p
```

```
ret void
```

```
}
```

; FIXME: This should use a single ADDI for the immediate.

```
define void @imm_store_i32_neg1(ptr %p) nounwind {
```

```
; RV64I-LABEL: imm_store_i32_neg1:
```

```
; RV64I:    # %bb.0:
```

```
; RV64I-NEXT:  li a1, -1
```

```
; RV64I-NEXT:
```

```
sw a1, 0(a0)
```

```
; RV64I-NEXT:  ret
```

```

;
; RV64IZBA-LABEL: imm_store_i32_neg1:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT: li a1, -1
; RV64IZBA-NEXT: sw a1, 0(a0)
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: imm_store_i32_neg1:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT: li a1, -1
; RV64IZBB-NEXT: sw a1, 0(a0)
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: imm_store_i32_neg1:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT: li a1, -1
; RV64IZBS-NEXT: sw a1, 0(a0)
; RV64IZBS-NEXT: ret
;
; RV64IXTHEADBB-LABEL: imm_store_i32_neg1:
; RV64IXTHEADBB: # %bb.0:
; RV64IXTHEADBB-NEXT: li a1, -1
; RV64IXTHEADBB-NEXT: sw a1, 0(a0)
; RV64IXTHEADBB-NEXT: ret
store i32 -1, ptr %p
ret void
}

```

```

define i64 @imm_5372288229() {
; RV64I-LABEL: imm_5372288229:
; RV64I:    # %bb.0:
; RV64I-NEXT: lui a0, 160
; RV64I-NEXT: addiw a0, a0, 437
; RV64I-NEXT: slli a0, a0, 13
; RV64I-NEXT: addi a0, a0, -795
; RV64I-NEXT: ret
;
; RV64IZBA-LABEL: imm_5372288229:
; RV64IZBA:
;   # %bb.0:
; RV64IZBA-NEXT: lui a0, 655797
; RV64IZBA-NEXT: slli.uw a0, a0, 1
; RV64IZBA-NEXT: addi a0, a0, -795
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: imm_5372288229:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT: lui a0, 160

```

```

; RV64IZBB-NEXT: addiw a0, a0, 437
; RV64IZBB-NEXT: slli a0, a0, 13
; RV64IZBB-NEXT: addi a0, a0, -795
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: imm_5372288229:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT: lui a0, 263018
; RV64IZBS-NEXT: addiw a0, a0, -795
; RV64IZBS-NEXT: bseti a0, a0, 32
; RV64IZBS-NEXT: ret
;
; RV64IXTHEADBB-LABEL: imm_5372288229:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT: lui a0, 160
; RV64IXTHEADBB-NEXT: addiw a0, a0, 437
; RV64IXTHEADBB-NEXT: slli a0, a0, 13
; RV64IXTHEADBB-NEXT: addi a0, a0, -795
; RV64IXTHEADBB-NEXT: ret
ret i64 5372288229
}

```

```

define i64 @imm_neg_5372288229() {
; RV64I-LABEL: imm_neg_5372288229:
; RV64I:    # %bb.0:
; RV64I-NEXT: lui a0, 1048416
; RV64I-NEXT:
    addiw a0, a0, -437
; RV64I-NEXT: slli a0, a0, 13
; RV64I-NEXT: addi a0, a0, 795
; RV64I-NEXT: ret
;
; RV64IZBA-LABEL: imm_neg_5372288229:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT: lui a0, 611378
; RV64IZBA-NEXT: addiw a0, a0, 265
; RV64IZBA-NEXT: sh1add a0, a0, a0
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: imm_neg_5372288229:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT: lui a0, 1048416
; RV64IZBB-NEXT: addiw a0, a0, -437
; RV64IZBB-NEXT: slli a0, a0, 13
; RV64IZBB-NEXT: addi a0, a0, 795
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: imm_neg_5372288229:

```

```

; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 785558
; RV64IZBS-NEXT:  addiw a0, a0, 795
; RV64IZBS-NEXT:  bclri a0, a0, 32
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_neg_5372288229:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 1048416
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -437
; RV64IXTHEADBB-NEXT:  slli a0, a0, 13
; RV64IXTHEADBB-NEXT:  addi a0, a0,
795
; RV64IXTHEADBB-NEXT:  ret
ret i64 -5372288229
}

```

```

define i64 @imm_8953813715() {
; RV64I-LABEL: imm_8953813715:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 267
; RV64I-NEXT:  addiw a0, a0, -637
; RV64I-NEXT:  slli a0, a0, 13
; RV64I-NEXT:  addi a0, a0, -1325
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_8953813715:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 437198
; RV64IZBA-NEXT:  addiw a0, a0, -265
; RV64IZBA-NEXT:  sh2add a0, a0, a0
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_8953813715:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 267
; RV64IZBB-NEXT:  addiw a0, a0, -637
; RV64IZBB-NEXT:  slli a0, a0, 13
; RV64IZBB-NEXT:  addi a0, a0, -1325
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_8953813715:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 88838
; RV64IZBS-NEXT:  addiw a0, a0, -1325
; RV64IZBS-NEXT:  bseti a0, a0, 33
; RV64IZBS-NEXT:  ret
;

```

```

; RV64IXTHEADBB-LABEL: imm_8953813715:
; RV64IXTHEADBB:    # %bb.0:
;
RV64IXTHEADBB-NEXT:  lui a0, 267
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -637
; RV64IXTHEADBB-NEXT:  slli a0, a0, 13
; RV64IXTHEADBB-NEXT:  addi a0, a0, -1325
; RV64IXTHEADBB-NEXT:  ret
ret i64 8953813715
}

define i64 @imm_neg_8953813715() {
; RV64I-LABEL: imm_neg_8953813715:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 1048309
; RV64I-NEXT:  addiw a0, a0, 637
; RV64I-NEXT:  slli a0, a0, 13
; RV64I-NEXT:  addi a0, a0, 1325
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_neg_8953813715:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 611378
; RV64IZBA-NEXT:  addiw a0, a0, 265
; RV64IZBA-NEXT:  sh2add a0, a0, a0
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_neg_8953813715:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 1048309
; RV64IZBB-NEXT:  addiw a0, a0, 637
; RV64IZBB-NEXT:  slli a0, a0, 13
; RV64IZBB-NEXT:  addi a0, a0, 1325
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_neg_8953813715:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:
    lui a0, 959738
; RV64IZBS-NEXT:  addiw a0, a0, 1325
; RV64IZBS-NEXT:  bclri a0, a0, 33
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_neg_8953813715:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 1048309
; RV64IXTHEADBB-NEXT:  addiw a0, a0, 637
; RV64IXTHEADBB-NEXT:  slli a0, a0, 13

```



```

; RV64IXTHEADBB-NEXT:  addi a0, a0, 1325
; RV64IXTHEADBB-NEXT:  ret
ret i64 -8953813715
}

define i64 @imm_16116864687() {
; RV64I-LABEL: imm_16116864687:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a0, 961
; RV64I-NEXT:  addiw a0, a0, -1475
; RV64I-NEXT:  slli a0, a0, 12
; RV64I-NEXT:  addi a0, a0, 1711
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_16116864687:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 437198
; RV64IZBA-NEXT:  addiw a0, a0, -265
; RV64IZBA-NEXT:  sh3add a0, a0, a0
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_16116864687:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 961
; RV64IZBB-NEXT:  addiw a0, a0, -1475
; RV64IZBB-NEXT:
;               slli a0, a0, 12
; RV64IZBB-NEXT:  addi a0, a0, 1711
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_16116864687:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 961
; RV64IZBS-NEXT:  addiw a0, a0, -1475
; RV64IZBS-NEXT:  slli a0, a0, 12
; RV64IZBS-NEXT:  addi a0, a0, 1711
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_16116864687:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 961
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -1475
; RV64IXTHEADBB-NEXT:  slli a0, a0, 12
; RV64IXTHEADBB-NEXT:  addi a0, a0, 1711
; RV64IXTHEADBB-NEXT:  ret
ret i64 16116864687
}

```

```

define i64 @imm_neg_16116864687() {
; RV64I-LABEL: imm_neg_16116864687:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 1047615
; RV64I-NEXT:  addiw a0, a0, 1475
; RV64I-NEXT:  slli a0, a0, 12
; RV64I-NEXT:  addi a0, a0, -1711
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_neg_16116864687:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 611378
; RV64IZBA-NEXT:
; RV64IZBA-NEXT:  addiw a0, a0, 265
; RV64IZBA-NEXT:  sh3add a0, a0, a0
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_neg_16116864687:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 1047615
; RV64IZBB-NEXT:  addiw a0, a0, 1475
; RV64IZBB-NEXT:  slli a0, a0, 12
; RV64IZBB-NEXT:  addi a0, a0, -1711
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_neg_16116864687:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 1047615
; RV64IZBS-NEXT:  addiw a0, a0, 1475
; RV64IZBS-NEXT:  slli a0, a0, 12
; RV64IZBS-NEXT:  addi a0, a0, -1711
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_neg_16116864687:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 1047615
; RV64IXTHEADBB-NEXT:  addiw a0, a0, 1475
; RV64IXTHEADBB-NEXT:  slli a0, a0, 12
; RV64IXTHEADBB-NEXT:  addi a0, a0, -1711
; RV64IXTHEADBB-NEXT:  ret
ret i64 -16116864687
}

```

```

define i64 @imm_2344336315() {
; RV64I-LABEL: imm_2344336315:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 143087
; RV64I-NEXT:

```

```

    slli a0, a0, 2
; RV64I-NEXT:  addi a0, a0, -1093
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_2344336315:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 143087
; RV64IZBA-NEXT:  slli a0, a0, 2
; RV64IZBA-NEXT:  addi a0, a0, -1093
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_2344336315:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 143087
; RV64IZBB-NEXT:  slli a0, a0, 2
; RV64IZBB-NEXT:  addi a0, a0, -1093
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_2344336315:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 143087
; RV64IZBS-NEXT:  slli a0, a0, 2
; RV64IZBS-NEXT:  addi a0, a0, -1093
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_2344336315:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 143087
; RV64IXTHEADBB-NEXT:  slli a0, a0, 2
; RV64IXTHEADBB-NEXT:  addi a0, a0, -1093
; RV64IXTHEADBB-NEXT:  ret
ret i64 2344336315 ; 0x8bbbbbbb
}

define i64 @imm_70370820078523() {
; RV64-NOPOOL-LABEL: imm_70370820078523:
;
RV64-NOPOOL:    # %bb.0:
; RV64-NOPOOL-NEXT:  lui a0, 256
; RV64-NOPOOL-NEXT:  addiw a0, a0, 31
; RV64-NOPOOL-NEXT:  slli a0, a0, 12
; RV64-NOPOOL-NEXT:  addi a0, a0, -273
; RV64-NOPOOL-NEXT:  slli a0, a0, 14
; RV64-NOPOOL-NEXT:  addi a0, a0, -1093
; RV64-NOPOOL-NEXT:  ret
;
; RV64I-POOL-LABEL: imm_70370820078523:
; RV64I-POOL:    # %bb.0:

```

```

; RV64I-POOL-NEXT: lui a0, %hi(.LCPI37_0)
; RV64I-POOL-NEXT: ld a0, %lo(.LCPI37_0)(a0)
; RV64I-POOL-NEXT: ret
;
; RV64IZBA-LABEL: imm_70370820078523:
; RV64IZBA: # %bb.0:
; RV64IZBA-NEXT: lui a0, 256
; RV64IZBA-NEXT: addiw a0, a0, 31
; RV64IZBA-NEXT: slli a0, a0, 12
; RV64IZBA-NEXT: addi a0, a0, -273
; RV64IZBA-NEXT: slli a0, a0, 14
; RV64IZBA-NEXT: addi a0, a0, -1093
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: imm_70370820078523:
; RV64IZBB: # %bb.0:
; RV64IZBB-NEXT: lui a0, 256
; RV64IZBB-NEXT: addiw a0, a0, 31
; RV64IZBB-NEXT: slli a0, a0, 12
; RV64IZBB-NEXT: addi a0, a0, -273
;
; RV64IZBB-NEXT: slli a0, a0, 14
; RV64IZBB-NEXT: addi a0, a0, -1093
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: imm_70370820078523:
; RV64IZBS: # %bb.0:
; RV64IZBS-NEXT: lui a0, 506812
; RV64IZBS-NEXT: addiw a0, a0, -1093
; RV64IZBS-NEXT: bseti a0, a0, 46
; RV64IZBS-NEXT: ret
;
; RV64IXTHEADBB-LABEL: imm_70370820078523:
; RV64IXTHEADBB: # %bb.0:
; RV64IXTHEADBB-NEXT: lui a0, 256
; RV64IXTHEADBB-NEXT: addiw a0, a0, 31
; RV64IXTHEADBB-NEXT: slli a0, a0, 12
; RV64IXTHEADBB-NEXT: addi a0, a0, -273
; RV64IXTHEADBB-NEXT: slli a0, a0, 14
; RV64IXTHEADBB-NEXT: addi a0, a0, -1093
; RV64IXTHEADBB-NEXT: ret
ret i64 70370820078523 ; 0x40007bbbbbbb
}

define i64 @imm_neg_9223372034778874949() {
; RV64I-LABEL: imm_neg_9223372034778874949:
; RV64I: # %bb.0:
; RV64I-NEXT: lui a0, 506812

```

```

; RV64I-NEXT:    addiw a0, a0, -1093
; RV64I-NEXT:    slli a1, a0, 63
; RV64I-NEXT:    add a0, a0, a1
; RV64I-NEXT:    ret
;
; RV64IZBA-LABEL: imm_neg_9223372034778874949:
;
RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:    lui a0, 506812
; RV64IZBA-NEXT:    addiw a0, a0, -1093
; RV64IZBA-NEXT:    slli a1, a0, 63
; RV64IZBA-NEXT:    add a0, a0, a1
; RV64IZBA-NEXT:    ret
;
; RV64IZBB-LABEL: imm_neg_9223372034778874949:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:    lui a0, 506812
; RV64IZBB-NEXT:    addiw a0, a0, -1093
; RV64IZBB-NEXT:    slli a1, a0, 63
; RV64IZBB-NEXT:    add a0, a0, a1
; RV64IZBB-NEXT:    ret
;
; RV64IZBS-LABEL: imm_neg_9223372034778874949:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:    lui a0, 506812
; RV64IZBS-NEXT:    addiw a0, a0, -1093
; RV64IZBS-NEXT:    bseti a0, a0, 63
; RV64IZBS-NEXT:    ret
;
; RV64IXTHEADBB-LABEL: imm_neg_9223372034778874949:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:    lui a0, 506812
; RV64IXTHEADBB-NEXT:    addiw a0, a0, -1093
; RV64IXTHEADBB-NEXT:    slli a1, a0, 63
; RV64IXTHEADBB-NEXT:    add a0, a0, a1
; RV64IXTHEADBB-NEXT:    ret
ret i64 -9223372034778874949 ; 0x800000007bbbbbbb
}

define
i64 @imm_neg_9223301666034697285() {
; RV64-NOPOOL-LABEL: imm_neg_9223301666034697285:
; RV64-NOPOOL:    # %bb.0:
; RV64-NOPOOL-NEXT:    lui a0, 917505
; RV64-NOPOOL-NEXT:    slli a0, a0, 8
; RV64-NOPOOL-NEXT:    addi a0, a0, 31
; RV64-NOPOOL-NEXT:    slli a0, a0, 12
; RV64-NOPOOL-NEXT:    addi a0, a0, -273

```

```

; RV64-NOPOOL-NEXT:  slli a0, a0, 14
; RV64-NOPOOL-NEXT:  addi a0, a0, -1093
; RV64-NOPOOL-NEXT:  ret
;
; RV64I-POOL-LABEL: imm_neg_9223301666034697285:
; RV64I-POOL:        # %bb.0:
; RV64I-POOL-NEXT:   lui a0, %hi(.LCPI39_0)
; RV64I-POOL-NEXT:   ld a0, %lo(.LCPI39_0)(a0)
; RV64I-POOL-NEXT:   ret
;
; RV64IZBA-LABEL: imm_neg_9223301666034697285:
; RV64IZBA:         # %bb.0:
; RV64IZBA-NEXT:    lui a0, 917505
; RV64IZBA-NEXT:    slli a0, a0, 8
; RV64IZBA-NEXT:    addi a0, a0, 31
; RV64IZBA-NEXT:    slli a0, a0, 12
; RV64IZBA-NEXT:    addi a0, a0, -273
; RV64IZBA-NEXT:    slli a0, a0, 14
; RV64IZBA-NEXT:    addi a0, a0, -1093
; RV64IZBA-NEXT:    ret
;
; RV64IZBB-LABEL:
imm_neg_9223301666034697285:
; RV64IZBB:         # %bb.0:
; RV64IZBB-NEXT:    lui a0, 917505
; RV64IZBB-NEXT:    slli a0, a0, 8
; RV64IZBB-NEXT:    addi a0, a0, 31
; RV64IZBB-NEXT:    slli a0, a0, 12
; RV64IZBB-NEXT:    addi a0, a0, -273
; RV64IZBB-NEXT:    slli a0, a0, 14
; RV64IZBB-NEXT:    addi a0, a0, -1093
; RV64IZBB-NEXT:    ret
;
; RV64IZBS-LABEL: imm_neg_9223301666034697285:
; RV64IZBS:         # %bb.0:
; RV64IZBS-NEXT:    lui a0, 506812
; RV64IZBS-NEXT:    addiw a0, a0, -1093
; RV64IZBS-NEXT:    bseti a0, a0, 46
; RV64IZBS-NEXT:    bseti a0, a0, 63
; RV64IZBS-NEXT:    ret
;
; RV64IXTHEADBB-LABEL: imm_neg_9223301666034697285:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 917505
; RV64IXTHEADBB-NEXT:  slli a0, a0, 8
; RV64IXTHEADBB-NEXT:  addi a0, a0, 31
; RV64IXTHEADBB-NEXT:  slli a0, a0, 12
; RV64IXTHEADBB-NEXT:  addi a0, a0, -273

```

```
; RV64IXTHEADBB-NEXT: slli a0, a0, 14
; RV64IXTHEADBB-NEXT: addi a0, a0, -1093
; RV64IXTHEADBB-NEXT: ret
ret i64 -9223301666034697285 ; 0x800040007bbbbbbb
}
```

```
define i64 @imm_neg_2219066437() {
; RV64I-LABEL: imm_neg_2219066437:
; RV64I:    # %bb.0:
; RV64I-NEXT: lui a0, 913135
; RV64I-NEXT: slli a0, a0, 2
; RV64I-NEXT: addi a0, a0, -1093
; RV64I-NEXT: ret
;
; RV64IZBA-LABEL: imm_neg_2219066437:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT: lui a0, 913135
; RV64IZBA-NEXT: slli a0, a0, 2
; RV64IZBA-NEXT: addi a0, a0, -1093
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: imm_neg_2219066437:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT: lui a0, 913135
; RV64IZBB-NEXT: slli a0, a0, 2
; RV64IZBB-NEXT: addi a0, a0, -1093
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: imm_neg_2219066437:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT: lui a0, 913135
; RV64IZBS-NEXT: slli a0, a0, 2
; RV64IZBS-NEXT: addi a0, a0, -1093
; RV64IZBS-NEXT: ret
;
; RV64IXTHEADBB-LABEL: imm_neg_2219066437:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT: lui a0, 913135
; RV64IXTHEADBB-NEXT:
    slli a0, a0, 2
; RV64IXTHEADBB-NEXT: addi a0, a0, -1093
; RV64IXTHEADBB-NEXT: ret
ret i64 -2219066437 ; 0xffffffff7bbbbbbb
}
```

```
define i64 @imm_neg_8798043653189() {
; RV64I-LABEL: imm_neg_8798043653189:
; RV64I:    # %bb.0:
```

```

; RV64I-NEXT:  lui a0, 917475
; RV64I-NEXT:  addiw a0, a0, -273
; RV64I-NEXT:  slli a0, a0, 14
; RV64I-NEXT:  addi a0, a0, -1093
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_neg_8798043653189:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 917475
; RV64IZBA-NEXT:  addiw a0, a0, -273
; RV64IZBA-NEXT:  slli a0, a0, 14
; RV64IZBA-NEXT:  addi a0, a0, -1093
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_neg_8798043653189:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 917475
; RV64IZBB-NEXT:  addiw a0, a0, -273
; RV64IZBB-NEXT:  slli a0, a0, 14
; RV64IZBB-NEXT:  addi a0, a0, -1093
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_neg_8798043653189:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 572348
; RV64IZBS-NEXT:
;               addiw a0, a0, -1093
; RV64IZBS-NEXT:  bclri a0, a0, 43
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_neg_8798043653189:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 917475
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -273
; RV64IXTHEADBB-NEXT:  slli a0, a0, 14
; RV64IXTHEADBB-NEXT:  addi a0, a0, -1093
; RV64IXTHEADBB-NEXT:  ret
;               ret i64 -8798043653189 ; 0xffff7ff8bbbbbbb
}

define i64 @imm_9223372034904144827() {
; RV64I-LABEL: imm_9223372034904144827:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 572348
; RV64I-NEXT:  addiw a0, a0, -1093
; RV64I-NEXT:  slli a1, a0, 63
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret

```



```

;
; RV64IZBA-LABEL: imm_9223372034904144827:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 572348
; RV64IZBA-NEXT:  addiw a0, a0, -1093
; RV64IZBA-NEXT:  slli a1, a0, 63
; RV64IZBA-NEXT:  add a0, a0, a1
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_9223372034904144827:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:
    lui a0, 572348
; RV64IZBB-NEXT:  addiw a0, a0, -1093
; RV64IZBB-NEXT:  slli a1, a0, 63
; RV64IZBB-NEXT:  add a0, a0, a1
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_9223372034904144827:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 572348
; RV64IZBS-NEXT:  addiw a0, a0, -1093
; RV64IZBS-NEXT:  bclri a0, a0, 63
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_9223372034904144827:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 572348
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -1093
; RV64IXTHEADBB-NEXT:  slli a1, a0, 63
; RV64IXTHEADBB-NEXT:  add a0, a0, a1
; RV64IXTHEADBB-NEXT:  ret
ret i64 9223372034904144827 ; 0x7fffffff8bbbbbbb
}

```

```

define i64 @imm_neg_9223354442718100411() {
; RV64-NOPOOL-LABEL: imm_neg_9223354442718100411:
; RV64-NOPOOL:    # %bb.0:
; RV64-NOPOOL-NEXT:  lui a0, 524287
; RV64-NOPOOL-NEXT:  slli a0, a0, 6
; RV64-NOPOOL-NEXT:  addi a0, a0, -29
; RV64-NOPOOL-NEXT:  slli a0, a0, 12
; RV64-NOPOOL-NEXT:
    addi a0, a0, -273
; RV64-NOPOOL-NEXT:  slli a0, a0, 14
; RV64-NOPOOL-NEXT:  addi a0, a0, -1093
; RV64-NOPOOL-NEXT:  ret
;
}

```

```

; RV64I-POOL-LABEL: imm_neg_9223354442718100411:
; RV64I-POOL:      # %bb.0:
; RV64I-POOL-NEXT: lui a0, %hi(.LCPI43_0)
; RV64I-POOL-NEXT: ld a0, %lo(.LCPI43_0)(a0)
; RV64I-POOL-NEXT: ret
;
; RV64IZBA-LABEL: imm_neg_9223354442718100411:
; RV64IZBA:      # %bb.0:
; RV64IZBA-NEXT: lui a0, 524287
; RV64IZBA-NEXT: slli a0, a0, 6
; RV64IZBA-NEXT: addi a0, a0, -29
; RV64IZBA-NEXT: slli a0, a0, 12
; RV64IZBA-NEXT: addi a0, a0, -273
; RV64IZBA-NEXT: slli a0, a0, 14
; RV64IZBA-NEXT: addi a0, a0, -1093
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: imm_neg_9223354442718100411:
; RV64IZBB:      # %bb.0:
; RV64IZBB-NEXT: lui a0, 524287
; RV64IZBB-NEXT: slli a0, a0, 6
; RV64IZBB-NEXT: addi a0, a0, -29
; RV64IZBB-NEXT: slli a0, a0, 12
; RV64IZBB-NEXT: addi a0, a0, -273
; RV64IZBB-NEXT: slli a0, a0, 14
; RV64IZBB-NEXT:
;         addi a0, a0, -1093
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: imm_neg_9223354442718100411:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT: lui a0, 572348
; RV64IZBS-NEXT: addiw a0, a0, -1093
; RV64IZBS-NEXT: bclri a0, a0, 44
; RV64IZBS-NEXT: bclri a0, a0, 63
; RV64IZBS-NEXT: ret
;
; RV64IXTHEADBB-LABEL: imm_neg_9223354442718100411:
; RV64IXTHEADBB: # %bb.0:
; RV64IXTHEADBB-NEXT: lui a0, 524287
; RV64IXTHEADBB-NEXT: slli a0, a0, 6
; RV64IXTHEADBB-NEXT: addi a0, a0, -29
; RV64IXTHEADBB-NEXT: slli a0, a0, 12
; RV64IXTHEADBB-NEXT: addi a0, a0, -273
; RV64IXTHEADBB-NEXT: slli a0, a0, 14
; RV64IXTHEADBB-NEXT: addi a0, a0, -1093
; RV64IXTHEADBB-NEXT: ret
ret i64 9223354442718100411 ; 0x7ffefff8bbbbbbb

```

```

}

define i64 @imm_2863311530() {
; RV64I-LABEL: imm_2863311530:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 349525
; RV64I-NEXT:  addiw a0, a0, 1365
; RV64I-NEXT:  slli a0, a0, 1
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_2863311530:
; RV64IZBA:
;   # %bb.0:
; RV64IZBA-NEXT:  lui a0, 349525
; RV64IZBA-NEXT:  addiw a0, a0, 1365
; RV64IZBA-NEXT:  slli a0, a0, 1
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_2863311530:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 349525
; RV64IZBB-NEXT:  addiw a0, a0, 1365
; RV64IZBB-NEXT:  slli a0, a0, 1
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_2863311530:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 349525
; RV64IZBS-NEXT:  addiw a0, a0, 1365
; RV64IZBS-NEXT:  slli a0, a0, 1
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_2863311530:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 349525
; RV64IXTHEADBB-NEXT:  addiw a0, a0, 1365
; RV64IXTHEADBB-NEXT:  slli a0, a0, 1
; RV64IXTHEADBB-NEXT:  ret
ret i64 2863311530 ; #0xaaaaaaaa
}

define i64 @imm_neg_2863311530() {
; RV64I-LABEL: imm_neg_2863311530:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 699051
; RV64I-NEXT:  addiw a0, a0, -1365
; RV64I-NEXT:  slli a0,

```

```

a0, 1
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_neg_2863311530:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 699051
; RV64IZBA-NEXT:  addiw a0, a0, -1365
; RV64IZBA-NEXT:  slli a0, a0, 1
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_neg_2863311530:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 699051
; RV64IZBB-NEXT:  addiw a0, a0, -1365
; RV64IZBB-NEXT:  slli a0, a0, 1
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_neg_2863311530:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 699051
; RV64IZBS-NEXT:  addiw a0, a0, -1365
; RV64IZBS-NEXT:  slli a0, a0, 1
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_neg_2863311530:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 699051
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -1365
; RV64IXTHEADBB-NEXT:  slli a0, a0, 1
; RV64IXTHEADBB-NEXT:  ret
ret i64 -2863311530 ; #0xffffffff55555556
}

```

```

define i64 @imm_2147486378() {
; RV64I-LABEL: imm_2147486378:
; RV64I:    # %bb.0:
; RV64I-NEXT:
    li a0, 1
; RV64I-NEXT:  slli a0, a0, 31
; RV64I-NEXT:  addi a0, a0, 1365
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_2147486378:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  li a0, 1
; RV64IZBA-NEXT:  slli a0, a0, 31
; RV64IZBA-NEXT:  addi a0, a0, 1365
; RV64IZBA-NEXT:  ret

```

```

;
; RV64IZBB-LABEL: imm_2147486378:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  li a0, 1
; RV64IZBB-NEXT:  slli a0, a0, 31
; RV64IZBB-NEXT:  addi a0, a0, 1365
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_2147486378:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  li a0, 1365
; RV64IZBS-NEXT:  bseti a0, a0, 31
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_2147486378:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  li a0, 1
; RV64IXTHEADBB-NEXT:  slli a0, a0, 31
; RV64IXTHEADBB-NEXT:  addi a0, a0, 1365
; RV64IXTHEADBB-NEXT:  ret
ret i64 2147485013
}

```

```

define i64 @imm_neg_2147485013() {
; RV64I-LABEL: imm_neg_2147485013:
; RV64I:    # %bb.0:
; RV64I-NEXT:
lui a0, 524288
; RV64I-NEXT:  addi a0, a0, -1365
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_neg_2147485013:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 524288
; RV64IZBA-NEXT:  addi a0, a0, -1365
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_neg_2147485013:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 524288
; RV64IZBB-NEXT:  addi a0, a0, -1365
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_neg_2147485013:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 524288
; RV64IZBS-NEXT:  addi a0, a0, -1365
; RV64IZBS-NEXT:  ret
}

```

```

;
; RV64IXTHEADBB-LABEL: imm_neg_2147485013:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 524288
; RV64IXTHEADBB-NEXT:  addi a0, a0, -1365
; RV64IXTHEADBB-NEXT:  ret
ret i64 -2147485013
}

```

```

define i64 @imm_12900924131259() {
; RV64I-LABEL: imm_12900924131259:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 188
; RV64I-NEXT:  addiw a0, a0, -1093
; RV64I-NEXT:  slli a0, a0, 24
; RV64I-NEXT:  addi a0,
a0, 1979
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_12900924131259:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 768955
; RV64IZBA-NEXT:  slli.uw a0, a0, 12
; RV64IZBA-NEXT:  addi a0, a0, 1979
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_12900924131259:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 188
; RV64IZBB-NEXT:  addiw a0, a0, -1093
; RV64IZBB-NEXT:  slli a0, a0, 24
; RV64IZBB-NEXT:  addi a0, a0, 1979
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_12900924131259:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 188
; RV64IZBS-NEXT:  addiw a0, a0, -1093
; RV64IZBS-NEXT:  slli a0, a0, 24
; RV64IZBS-NEXT:  addi a0, a0, 1979
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_12900924131259:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 188
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -1093
; RV64IXTHEADBB-NEXT:  slli a0, a0, 24
; RV64IXTHEADBB-NEXT:  addi a0, a0, 1979

```

```

; RV64IXTHEADBB-NEXT:  ret
ret i64 12900924131259
}

define
i64 @imm_50394234880() {
; RV64I-LABEL: imm_50394234880:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a0, 188
; RV64I-NEXT:  addiw a0, a0, -1093
; RV64I-NEXT:  slli a0, a0, 16
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_50394234880:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 768955
; RV64IZBA-NEXT:  slli.uw a0, a0, 4
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_50394234880:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 188
; RV64IZBB-NEXT:  addiw a0, a0, -1093
; RV64IZBB-NEXT:  slli a0, a0, 16
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_50394234880:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 188
; RV64IZBS-NEXT:  addiw a0, a0, -1093
; RV64IZBS-NEXT:  slli a0, a0, 16
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_50394234880:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 188
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -1093
; RV64IXTHEADBB-NEXT:  slli a0, a0, 16
; RV64IXTHEADBB-NEXT:  ret
ret
i64 50394234880
}

define i64 @imm_12900936431479() {
; RV64I-LABEL: imm_12900936431479:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a0, 192239
; RV64I-NEXT:  slli a0, a0, 2

```

```

; RV64I-NEXT:  addi a0, a0, -1093
; RV64I-NEXT:  slli a0, a0, 12
; RV64I-NEXT:  addi a0, a0, 1911
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_12900936431479:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 768956
; RV64IZBA-NEXT:  addi a0, a0, -1093
; RV64IZBA-NEXT:  slli.uw a0, a0, 12
; RV64IZBA-NEXT:  addi a0, a0, 1911
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_12900936431479:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 192239
; RV64IZBB-NEXT:  slli a0, a0, 2
; RV64IZBB-NEXT:  addi a0, a0, -1093
; RV64IZBB-NEXT:  slli a0, a0, 12
; RV64IZBB-NEXT:  addi a0, a0, 1911
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_12900936431479:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 192239
; RV64IZBS-NEXT:  slli a0, a0, 2
; RV64IZBS-NEXT:  addi a0, a0, -1093
; RV64IZBS-NEXT:
;               slli a0, a0, 12
; RV64IZBS-NEXT:  addi a0, a0, 1911
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_12900936431479:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 192239
; RV64IXTHEADBB-NEXT:  slli a0, a0, 2
; RV64IXTHEADBB-NEXT:  addi a0, a0, -1093
; RV64IXTHEADBB-NEXT:  slli a0, a0, 12
; RV64IXTHEADBB-NEXT:  addi a0, a0, 1911
; RV64IXTHEADBB-NEXT:  ret
ret i64 12900936431479
}

define i64 @imm_12900918536874() {
; RV64I-LABEL: imm_12900918536874:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 384477
; RV64I-NEXT:  addiw a0, a0, 1365

```



```

; RV64I-NEXT:  slli a0, a0, 12
; RV64I-NEXT:  addi a0, a0, 1365
; RV64I-NEXT:  slli a0, a0, 1
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_12900918536874:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 768955
; RV64IZBA-NEXT:  addi a0, a0, -1365
; RV64IZBA-NEXT:  slli.uw a0, a0, 12
; RV64IZBA-NEXT:  addi a0, a0, -1366
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_12900918536874:
;
RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 384477
; RV64IZBB-NEXT:  addiw a0, a0, 1365
; RV64IZBB-NEXT:  slli a0, a0, 12
; RV64IZBB-NEXT:  addi a0, a0, 1365
; RV64IZBB-NEXT:  slli a0, a0, 1
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_12900918536874:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 384477
; RV64IZBS-NEXT:  addiw a0, a0, 1365
; RV64IZBS-NEXT:  slli a0, a0, 12
; RV64IZBS-NEXT:  addi a0, a0, 1365
; RV64IZBS-NEXT:  slli a0, a0, 1
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_12900918536874:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 384477
; RV64IXTHEADBB-NEXT:  addiw a0, a0, 1365
; RV64IXTHEADBB-NEXT:  slli a0, a0, 12
; RV64IXTHEADBB-NEXT:  addi a0, a0, 1365
; RV64IXTHEADBB-NEXT:  slli a0, a0, 1
; RV64IXTHEADBB-NEXT:  ret
ret i64 12900918536874
}

define i64 @imm_12900925247761() {
; RV64I-LABEL: imm_12900925247761:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 384478
; RV64I-NEXT:  addiw

```

```

a0, a0, -1911
; RV64I-NEXT:  slli a0, a0, 13
; RV64I-NEXT:  addi a0, a0, -2048
; RV64I-NEXT:  addi a0, a0, -1775
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_12900925247761:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 768955
; RV64IZBA-NEXT:  addi a0, a0, 273
; RV64IZBA-NEXT:  slli.uw a0, a0, 12
; RV64IZBA-NEXT:  addi a0, a0, 273
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_12900925247761:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 384478
; RV64IZBB-NEXT:  addiw a0, a0, -1911
; RV64IZBB-NEXT:  slli a0, a0, 13
; RV64IZBB-NEXT:  addi a0, a0, -2048
; RV64IZBB-NEXT:  addi a0, a0, -1775
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_12900925247761:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 384478
; RV64IZBS-NEXT:  addiw a0, a0, -1911
; RV64IZBS-NEXT:  slli a0, a0, 13
; RV64IZBS-NEXT:  addi a0, a0, -2048
; RV64IZBS-NEXT:  addi a0, a0, -1775
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_12900925247761:
; RV64IXTHEADBB:
;   # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 384478
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -1911
; RV64IXTHEADBB-NEXT:  slli a0, a0, 13
; RV64IXTHEADBB-NEXT:  addi a0, a0, -2048
; RV64IXTHEADBB-NEXT:  addi a0, a0, -1775
; RV64IXTHEADBB-NEXT:  ret
ret i64 12900925247761
}

define i64 @imm_7158272001() {
; RV64I-LABEL: imm_7158272001:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 427

```

```

; RV64I-NEXT:    addiw a0, a0, -1367
; RV64I-NEXT:    slli a0, a0, 12
; RV64I-NEXT:    addi a0, a0, 1
; RV64I-NEXT:    ret
;
; RV64IZBA-LABEL: imm_7158272001:
; RV64IZBA:      # %bb.0:
; RV64IZBA-NEXT:    lui a0, 349525
; RV64IZBA-NEXT:    sh2add a0, a0, a0
; RV64IZBA-NEXT:    addi a0, a0, 1
; RV64IZBA-NEXT:    ret
;
; RV64IZBB-LABEL: imm_7158272001:
; RV64IZBB:      # %bb.0:
; RV64IZBB-NEXT:    lui a0, 427
; RV64IZBB-NEXT:    addiw a0, a0, -1367
; RV64IZBB-NEXT:    slli a0, a0, 12
; RV64IZBB-NEXT:    addi a0, a0, 1
; RV64IZBB-NEXT:    ret
;
; RV64IZBS-LABEL: imm_7158272001:
; RV64IZBS:
;   # %bb.0:
; RV64IZBS-NEXT:    lui a0, 427
; RV64IZBS-NEXT:    addiw a0, a0, -1367
; RV64IZBS-NEXT:    slli a0, a0, 12
; RV64IZBS-NEXT:    addi a0, a0, 1
; RV64IZBS-NEXT:    ret
;
; RV64IXTHEADBB-LABEL: imm_7158272001:
; RV64IXTHEADBB:   # %bb.0:
; RV64IXTHEADBB-NEXT:    lui a0, 427
; RV64IXTHEADBB-NEXT:    addiw a0, a0, -1367
; RV64IXTHEADBB-NEXT:    slli a0, a0, 12
; RV64IXTHEADBB-NEXT:    addi a0, a0, 1
; RV64IXTHEADBB-NEXT:    ret
ret i64 7158272001 ; 0x0000_0001_aaaa_9001
}

```

```

define i64 @imm_12884889601() {
; RV64I-LABEL: imm_12884889601:
; RV64I:      # %bb.0:
; RV64I-NEXT:    lui a0, 768
; RV64I-NEXT:    addiw a0, a0, -3
; RV64I-NEXT:    slli a0, a0, 12
; RV64I-NEXT:    addi a0, a0, 1
; RV64I-NEXT:    ret
;

```

```

; RV64IZBA-LABEL: imm_12884889601:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 349525
; RV64IZBA-NEXT:  sh3add a0, a0, a0
; RV64IZBA-NEXT:  addi a0, a0, 1
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_12884889601:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:
    lui a0, 768
; RV64IZBB-NEXT:  addiw a0, a0, -3
; RV64IZBB-NEXT:  slli a0, a0, 12
; RV64IZBB-NEXT:  addi a0, a0, 1
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_12884889601:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 768
; RV64IZBS-NEXT:  addiw a0, a0, -3
; RV64IZBS-NEXT:  slli a0, a0, 12
; RV64IZBS-NEXT:  addi a0, a0, 1
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_12884889601:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 768
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -3
; RV64IXTHEADBB-NEXT:  slli a0, a0, 12
; RV64IXTHEADBB-NEXT:  addi a0, a0, 1
; RV64IXTHEADBB-NEXT:  ret
ret i64 12884889601 ; 0x0000_0002_ffff_d001
}

```

```

define i64 @imm_neg_3435982847() {
; RV64I-LABEL: imm_neg_3435982847:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 1048371
; RV64I-NEXT:  addiw a0, a0, 817
; RV64I-NEXT:  slli a0, a0, 12
; RV64I-NEXT:  addi a0, a0, 1
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_neg_3435982847:
; RV64IZBA:
    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 768955
; RV64IZBA-NEXT:  sh1add a0, a0, a0

```

```

; RV64IZBA-NEXT:  addi a0, a0, 1
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_neg_3435982847:
; RV64IZBB:      # %bb.0:
; RV64IZBB-NEXT:  lui a0, 1048371
; RV64IZBB-NEXT:  addiw a0, a0, 817
; RV64IZBB-NEXT:  slli a0, a0, 12
; RV64IZBB-NEXT:  addi a0, a0, 1
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_neg_3435982847:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT:  lui a0, 734001
; RV64IZBS-NEXT:  addiw a0, a0, 1
; RV64IZBS-NEXT:  bclri a0, a0, 31
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_neg_3435982847:
; RV64IXTHEADBB: # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 1048371
; RV64IXTHEADBB-NEXT:  addiw a0, a0, 817
; RV64IXTHEADBB-NEXT:  slli a0, a0, 12
; RV64IXTHEADBB-NEXT:  addi a0, a0, 1
; RV64IXTHEADBB-NEXT:  ret
ret i64 -3435982847 ; 0xffff_ffff_3333_1001
}

```

```

define i64 @imm_neg_5726842879() {
; RV64I-LABEL: imm_neg_5726842879:
; RV64I:      # %bb.0:
; RV64I-NEXT:
    lui a0, 1048235
; RV64I-NEXT:  addiw a0, a0, -1419
; RV64I-NEXT:  slli a0, a0, 12
; RV64I-NEXT:  addi a0, a0, 1
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_neg_5726842879:
; RV64IZBA:   # %bb.0:
; RV64IZBA-NEXT:  lui a0, 768945
; RV64IZBA-NEXT:  sh2add a0, a0, a0
; RV64IZBA-NEXT:  addi a0, a0, 1
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_neg_5726842879:
; RV64IZBB:   # %bb.0:
; RV64IZBB-NEXT:  lui a0, 1048235

```

```

; RV64IZBB-NEXT: addiw a0, a0, -1419
; RV64IZBB-NEXT: slli a0, a0, 12
; RV64IZBB-NEXT: addi a0, a0, 1
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: imm_neg_5726842879:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT: lui a0, 698997
; RV64IZBS-NEXT: addiw a0, a0, 1
; RV64IZBS-NEXT: bclri a0, a0, 32
; RV64IZBS-NEXT: ret
;
; RV64IXTHEADBB-LABEL: imm_neg_5726842879:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT: lui a0, 1048235
; RV64IXTHEADBB-NEXT: addiw a0, a0, -1419
; RV64IXTHEADBB-NEXT: slli a0, a0, 12
; RV64IXTHEADBB-NEXT:
    addi a0, a0, 1
; RV64IXTHEADBB-NEXT: ret
ret i64 -5726842879 ; 0xffff_fffe_aaa7_5001
}

```

```

define i64 @imm_neg_10307948543() {
; RV64I-LABEL: imm_neg_10307948543:
; RV64I:    # %bb.0:
; RV64I-NEXT: lui a0, 1047962
; RV64I-NEXT: addiw a0, a0, -1645
; RV64I-NEXT: slli a0, a0, 12
; RV64I-NEXT: addi a0, a0, 1
; RV64I-NEXT: ret
;
; RV64IZBA-LABEL: imm_neg_10307948543:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT: lui a0, 768955
; RV64IZBA-NEXT: sh3add a0, a0, a0
; RV64IZBA-NEXT: addi a0, a0, 1
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: imm_neg_10307948543:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT: lui a0, 1047962
; RV64IZBB-NEXT: addiw a0, a0, -1645
; RV64IZBB-NEXT: slli a0, a0, 12
; RV64IZBB-NEXT: addi a0, a0, 1
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: imm_neg_10307948543:

```

```

; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 629139
; RV64IZBS-NEXT:  addiw a0, a0, 1
; RV64IZBS-NEXT:  bclri a0, a0, 33
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL:
imm_neg_10307948543:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 1047962
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -1645
; RV64IXTHEADBB-NEXT:  slli a0, a0, 12
; RV64IXTHEADBB-NEXT:  addi a0, a0, 1
; RV64IXTHEADBB-NEXT:  ret
ret i64 -10307948543 ; 0xffff_fffd_9999_3001
}

```

```

define i64 @li_ropi_1() {
; RV64I-LABEL: li_ropi_1:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a0, -17
; RV64I-NEXT:  slli a0, a0, 43
; RV64I-NEXT:  addi a0, a0, -1
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: li_ropi_1:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  li a0, -17
; RV64IZBA-NEXT:  slli a0, a0, 43
; RV64IZBA-NEXT:  addi a0, a0, -1
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: li_ropi_1:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  li a0, -18
; RV64IZBB-NEXT:  ropi a0, a0, 21
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: li_ropi_1:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  li a0, -17
; RV64IZBS-NEXT:  slli a0, a0, 43
; RV64IZBS-NEXT:  addi a0, a0,
-1
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: li_ropi_1:
; RV64IXTHEADBB:    # %bb.0:

```

```
; RV64IXTHEADBB-NEXT: li a0, -18
; RV64IXTHEADBB-NEXT: th.srri a0, a0, 21
; RV64IXTHEADBB-NEXT: ret
ret i64 -149533581377537
}
```

```
define i64 @li_ropi_2() {
; RV64I-LABEL: li_ropi_2:
; RV64I:    # %bb.0:
; RV64I-NEXT: li a0, -5
; RV64I-NEXT: slli a0, a0, 60
; RV64I-NEXT: addi a0, a0, -6
; RV64I-NEXT: ret
;
; RV64IZBA-LABEL: li_ropi_2:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT: li a0, -5
; RV64IZBA-NEXT: slli a0, a0, 60
; RV64IZBA-NEXT: addi a0, a0, -6
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: li_ropi_2:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT: li a0, -86
; RV64IZBB-NEXT: rori a0, a0, 4
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: li_ropi_2:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT: li a0, -5
; RV64IZBS-NEXT: slli a0, a0, 60
; RV64IZBS-NEXT: addi a0, a0, -6
; RV64IZBS-NEXT: ret
;
; RV64IXTHEADBB-LABEL: li_ropi_2:
; RV64IXTHEADBB:
;   # %bb.0:
; RV64IXTHEADBB-NEXT: li a0, -86
; RV64IXTHEADBB-NEXT: th.srri a0, a0, 4
; RV64IXTHEADBB-NEXT: ret
ret i64 -5764607523034234886
}
```

```
define i64 @li_ropi_3() {
; RV64I-LABEL: li_ropi_3:
; RV64I:    # %bb.0:
; RV64I-NEXT: li a0, -17
; RV64I-NEXT: slli a0, a0, 27
```



```

; RV64I-NEXT:  addi a0, a0, -1
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: li_ropi_3:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  li a0, -17
; RV64IZBA-NEXT:  slli a0, a0, 27
; RV64IZBA-NEXT:  addi a0, a0, -1
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: li_ropi_3:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  li a0, -18
; RV64IZBB-NEXT:  ropi a0, a0, 37
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: li_ropi_3:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  li a0, -17
; RV64IZBS-NEXT:  slli a0, a0, 27
; RV64IZBS-NEXT:  addi a0, a0, -1
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: li_ropi_3:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  li a0, -18
; RV64IXTHEADBB-NEXT:  th.srri a0,
a0, 37
; RV64IXTHEADBB-NEXT:  ret
ret i64 -2281701377
}

```

; This used to assert when compiled with Zba.

```

define i64 @PR54812() {
; RV64I-LABEL: PR54812:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 1048447
; RV64I-NEXT:  addiw a0, a0, 1407
; RV64I-NEXT:  slli a0, a0, 12
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: PR54812:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 872917
; RV64IZBA-NEXT:  sh1add a0, a0, a0
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: PR54812:

```

```

; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 1048447
; RV64IZBB-NEXT:  addiw a0, a0, 1407
; RV64IZBB-NEXT:  slli a0, a0, 12
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: PR54812:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 1045887
; RV64IZBS-NEXT:  bclri a0, a0, 31
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: PR54812:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 1048447
; RV64IXTHEADBB-NEXT:  addiw a0, a0, 1407
; RV64IXTHEADBB-NEXT:  slli a0, a0, 12
; RV64IXTHEADBB-NEXT:
    ret
ret i64 -2158497792;
}

```

```

define signext i32 @pos_2048() nounwind {

```

```

; RV64I-LABEL: pos_2048:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a0, 1
; RV64I-NEXT:  slli a0, a0, 11
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: pos_2048:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  li a0, 1
; RV64IZBA-NEXT:  slli a0, a0, 11
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: pos_2048:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  li a0, 1
; RV64IZBB-NEXT:  slli a0, a0, 11
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: pos_2048:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  bseti a0, zero, 11
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: pos_2048:
; RV64IXTHEADBB:    # %bb.0:

```

```
; RV64IXTHEADBB-NEXT: li a0, 1
; RV64IXTHEADBB-NEXT: slli a0, a0, 11
; RV64IXTHEADBB-NEXT: ret
ret i32 2048
}
```

```
define i64 @imm64_same_lo_hi() nounwind {
; RV64I-LABEL: imm64_same_lo_hi:
; RV64I:    # %bb.0:
; RV64I-NEXT: lui a0, 65793
; RV64I-NEXT: addiw a0, a0, 16
; RV64I-NEXT:
    slli a1, a0, 32
; RV64I-NEXT: add a0, a0, a1
; RV64I-NEXT: ret
;
; RV64IZBA-LABEL: imm64_same_lo_hi:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT: lui a0, 65793
; RV64IZBA-NEXT: addiw a0, a0, 16
; RV64IZBA-NEXT: slli a1, a0, 32
; RV64IZBA-NEXT: add a0, a0, a1
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: imm64_same_lo_hi:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT: lui a0, 65793
; RV64IZBB-NEXT: addiw a0, a0, 16
; RV64IZBB-NEXT: slli a1, a0, 32
; RV64IZBB-NEXT: add a0, a0, a1
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: imm64_same_lo_hi:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT: lui a0, 65793
; RV64IZBS-NEXT: addiw a0, a0, 16
; RV64IZBS-NEXT: slli a1, a0, 32
; RV64IZBS-NEXT: add a0, a0, a1
; RV64IZBS-NEXT: ret
;
; RV64IXTHEADBB-LABEL: imm64_same_lo_hi:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT: lui a0, 65793
; RV64IXTHEADBB-NEXT: addiw a0, a0, 16
; RV64IXTHEADBB-NEXT: slli a1, a0, 32
; RV64IXTHEADBB-NEXT: add a0, a0, a1
;
}
```

```

RV64IXTHEADBB-NEXT:  ret
ret i64 1157442765409226768 ; 0x0101010101010101
}

```

; Same as above with optsize. Make sure we use constant pool on RV64

```
define i64 @imm64_same_lo_hi_optsize() nounwind optsize {
```

```
; RV64-NOPOOL-LABEL: imm64_same_lo_hi_optsize:
```

```
; RV64-NOPOOL:      # %bb.0:
```

```
; RV64-NOPOOL-NEXT:  lui a0, 65793
```

```
; RV64-NOPOOL-NEXT:  addiw a0, a0, 16
```

```
; RV64-NOPOOL-NEXT:  slli a1, a0, 32
```

```
; RV64-NOPOOL-NEXT:  add a0, a0, a1
```

```
; RV64-NOPOOL-NEXT:  ret
```

```
;
```

```
; RV64I-POOL-LABEL: imm64_same_lo_hi_optsize:
```

```
; RV64I-POOL:      # %bb.0:
```

```
; RV64I-POOL-NEXT:  lui a0, %hi(.LCPI64_0)
```

```
; RV64I-POOL-NEXT:  ld a0, %lo(.LCPI64_0)(a0)
```

```
; RV64I-POOL-NEXT:  ret
```

```
;
```

```
; RV64IZBA-LABEL: imm64_same_lo_hi_optsize:
```

```
; RV64IZBA:      # %bb.0:
```

```
; RV64IZBA-NEXT:  lui a0, 65793
```

```
; RV64IZBA-NEXT:  addiw a0, a0, 16
```

```
; RV64IZBA-NEXT:  slli a1, a0, 32
```

```
; RV64IZBA-NEXT:  add a0, a0, a1
```

```
; RV64IZBA-NEXT:  ret
```

```
;
```

```
; RV64IZBB-LABEL: imm64_same_lo_hi_optsize:
```

```
; RV64IZBB:      # %bb.0:
```

```
; RV64IZBB-NEXT:  lui
a0, 65793
```

```
; RV64IZBB-NEXT:  addiw a0, a0, 16
```

```
; RV64IZBB-NEXT:  slli a1, a0, 32
```

```
; RV64IZBB-NEXT:  add a0, a0, a1
```

```
; RV64IZBB-NEXT:  ret
```

```
;
```

```
; RV64IZBS-LABEL: imm64_same_lo_hi_optsize:
```

```
; RV64IZBS:      # %bb.0:
```

```
; RV64IZBS-NEXT:  lui a0, 65793
```

```
; RV64IZBS-NEXT:  addiw a0, a0, 16
```

```
; RV64IZBS-NEXT:  slli a1, a0, 32
```

```
; RV64IZBS-NEXT:  add a0, a0, a1
```

```
; RV64IZBS-NEXT:  ret
```

```
;
```

```
; RV64IXTHEADBB-LABEL: imm64_same_lo_hi_optsize:
```

```
; RV64IXTHEADBB:      # %bb.0:
```

```
; RV64IXTHEADBB-NEXT:  lui a0, 65793
```

```
; RV64IXTHEADBB-NEXT:  addiw a0, a0, 16
; RV64IXTHEADBB-NEXT:  slli a1, a0, 32
; RV64IXTHEADBB-NEXT:  add a0, a0, a1
; RV64IXTHEADBB-NEXT:  ret
ret i64 1157442765409226768 ; 0x0101010101010101
}
```

Copyright (c) 2006 Kirill Simonov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

\$OpenBSD: COPYRIGHT,v 1.3 2003/06/02 20:18:36 millert Exp \$

Copyright 1992, 1993, 1994 Henry Spencer. All rights reserved.  
This software is not subject to any license of the American Telephone and Telegraph Company or of the Regents of the University of California.

Permission is granted to anyone to use this software for any purpose on any computer system, and to alter it and redistribute it, subject to the following restrictions:

1. The author is not responsible for the consequences of use of this software, no matter how awful, even if they arise from flaws in it.
2. The origin of this software must not be misrepresented, either by explicit claim or by omission. Since few users ever read sources, credits must appear in the documentation.
3. Altered versions must be plainly marked as such, and must not be misrepresented as being the original software. Since few users ever read sources, credits must appear in the documentation.
4. This notice may not be removed or

altered.

=====

/\*\_

\* Copyright (c) 1994

\* The Regents of the University of California. All rights reserved.

\*

\* Redistribution and use in source and binary forms, with or without

\* modification, are permitted provided that the following conditions

\* are met:

\* 1. Redistributions of source code must retain the above copyright

\* notice, this list of conditions and the following disclaimer.

\* 2. Redistributions in binary form must reproduce the above copyright

\* notice, this list of conditions and the following disclaimer in the

\* documentation and/or other materials provided with the distribution.

\* 3. Neither the name of the University nor the names of its contributors

\* may be used to endorse or promote products derived from this software

\* without specific prior written permission.

\*

\* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND

\* ANY EXPRESS OR IMPLIED WARRANTIES,

INCLUDING, BUT NOT LIMITED TO, THE

\* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

\* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE

\* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

\* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS

\* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

\* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT

\* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY

\* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF

\* SUCH DAMAGE.

\*

\* @(#)COPYRIGHT 8.1 (Berkeley) 3/16/94

\*/

Copyright 2008, Google Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions are  
met:

\* Redistributions of source code must retain the above copyright  
notice, this list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above  
copyright notice, this list of conditions and the following disclaimer  
in the documentation and/or other materials provided with the  
distribution.

\* Neither the name of Google Inc. nor the names of its

contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR

CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mcpu=skylake-avx512 -mattr=prefer-256-bit | FileCheck %s
--check-prefixes=CHECK,CHECK-SKX
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mcpu=skylake-avx512 -mattr=prefer-256-bit,avx512vbmi |
FileCheck %s --check-prefixes=CHECK,CHECK-SKX,CHECK-SKX-VBMI
; Make sure CPUs default to prefer-256-bit. avx512vnni isn't interesting as it just adds an isel peephole for
vpmaddwd+vpadd
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mcpu=skylake-avx512 | FileCheck %s --check-
prefixes=CHECK,CHECK-AVX512
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mattr=-avx512vnni -mcpu=cascadelake | FileCheck %s --
check-prefixes=CHECK,CHECK-AVX512
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mattr=-avx512vnni -mcpu=cooperlake | FileCheck %s --
check-prefixes=CHECK,CHECK-AVX512
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mcpu=cannonlake | FileCheck %s --check-
prefixes=CHECK,CHECK-VBMI
;
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mattr=-avx512vnni -mcpu=icelake-client | FileCheck %s --
check-prefixes=CHECK,CHECK-VBMI
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mattr=-avx512vnni -mcpu=icelake-server | FileCheck %s --
check-prefixes=CHECK,CHECK-VBMI
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mattr=-avx512vnni -mcpu=tigerlake | FileCheck %s --check-
prefixes=CHECK,CHECK-VBMI
```

; This file primarily contains tests for specific places in X86ISelLowering.cpp that needed be made aware of the legalizer not allowing 512-bit vectors due to prefer-256-bit even though AVX512 is enabled.

```
define dso_local void @add256(<16 x i32>* %a, <16 x i32>* %b, <16 x i32>* %c) "min-legal-vector-
width"="256" {
; CHECK-LABEL: add256:
; CHECK:      # %bb.0:
; CHECK-NEXT:  vmovdqa (%rdi), %ymm0
; CHECK-NEXT:  vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT:  vpadd 32(%rsi), %ymm1, %ymm1
```

```

; CHECK-NEXT:    vpaddd (%rsi), %ymm0, %ymm0
; CHECK-NEXT:    vmovdqa %ymm0, (%rdx)
; CHECK-NEXT:
    vmovdqa %ymm1, 32(%rdx)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq

%d = load <16 x i32>, <16 x i32>* %a
%e = load <16 x i32>, <16 x i32>* %b
%f = add <16 x i32> %d, %e
store <16 x i32> %f, <16 x i32>* %c
ret void
}

```

```
; CHECK-LABEL: add512:
; CHECK:      # %bb.0:
; CHECK-NEXT:  vmovdqa64 (%rdi), %zmm0
; CHECK-NEXT:  vpaddd (%rsi), %zmm0, %zmm0
; CHECK-NEXT:  vmovdqa64 %zmm0, (%rdx)
; CHECK-NEXT:  vzeroupper
; CHECK-NEXT:  retq

%d = load <16 x i32>, <16 x i32>* %a
%e = load <16 x i32>, <16 x i32>* %b
%f = add <16 x i32> %d, %e
store <16 x i32> %f, <16 x i32>* %c
ret void
}
```

[illegible]



[illegible]

```
define dso_local void @pmaddwd_32_256(<32 x i16>* %APtr, <32 x i16>* %BPtr, <16 x i32>* %CPtr) "min-legal-vector-width"="256" {
; CHECK-LABEL: pmaddwd_32_256:
; CHECK:      # %bb.0:
; CHECK-NEXT:  vmovdqa (%rdi), %ymm0
; CHECK-NEXT:  vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT:  vpmaddwd 32(%rsi), %ymm1, %ymm1
```

```

; CHECK-NEXT: vpmaddwd (%rsi), %ymm0, %ymm0
; CHECK-NEXT: vmovdqa %ymm0, (%rdx)
; CHECK-NEXT: vmovdqa %ymm1, 32(%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq

%A = load <32 x i16>, <32 x i16>* %APtr
%B = load <32 x i16>, <32 x i16>* %BPtr
%a = sext <32 x i16> %A to <32 x i32>
%b = sext <32 x i16> %B to <32 x i32>
%m = mul nsw <32 x i32> %a, %b
%odd = shufflevector <32 x i32> %m, <32 x i32> undef, <16 x i32> <i32 0, i32 2, i32 4, i32 6, i32 8, i32 10, i32
12, i32 14, i32 16, i32 18, i32 20, i32 22, i32 24, i32 26, i32 28, i32 30>
%even = shufflevector <32 x i32> %m, <32 x i32> undef, <16 x i32> <i32 1, i32 3, i32 5, i32
7, i32 9, i32 11, i32 13, i32 15, i32 17, i32 19, i32 21, i32 23, i32 25, i32 27, i32 29, i32 31>
%ret = add <16 x i32> %odd, %even
store <16 x i32> %ret, <16 x i32>* %CPtr
ret void
}

define dso_local void @pmaddwd_32_512(<32 x i16>* %APtr, <32 x i16>* %BPtr, <16 x i32>* %CPtr) "min-
legal-vector-width"="512" {
; CHECK-LABEL: pmaddwd_32_512:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa64 (%rdi), %zmm0
; CHECK-NEXT: vpmaddwd (%rsi), %zmm0, %zmm0
; CHECK-NEXT: vmovdqa64 %zmm0, (%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq

%A = load <32 x i16>, <32 x i16>* %APtr
%B = load <32 x i16>, <32 x i16>* %BPtr
%a = sext <32 x i16> %A to <32 x i32>
%b = sext <32 x i16> %B to <32 x i32>
%m = mul nsw <32 x i32> %a, %b
%odd = shufflevector <32 x i32> %m, <32 x i32> undef, <16 x i32> <i32 0, i32 2, i32 4, i32 6, i32 8, i32 10, i32
12, i32 14, i32 16, i32 18, i32 20, i32 22, i32 24, i32 26, i32 28, i32 30>
%even = shufflevector <32 x i32> %m, <32 x i32>
undef, <16 x i32> <i32 1, i32 3, i32 5, i32 7, i32 9, i32 11, i32 13, i32 15, i32 17, i32 19, i32 21, i32 23, i32 25, i32
27, i32 29, i32 31>
%ret = add <16 x i32> %odd, %even
store <16 x i32> %ret, <16 x i32>* %CPtr
ret void
}

define dso_local void @psubus_64i8_max_256(<64 x i8>* %xptr, <64 x i8>* %yptr, <64 x i8>* %zptr) "min-legal-
vector-width"="256" {
; CHECK-LABEL: psubus_64i8_max_256:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rdi), %ymm0

```

```

; CHECK-NEXT: vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT: vpsubusb 32(%rsi), %ymm1, %ymm1
; CHECK-NEXT: vpsubusb (%rsi), %ymm0, %ymm0
; CHECK-NEXT: vmovdqa %ymm0, (%rdx)
; CHECK-NEXT: vmovdqa %ymm1, 32(%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%x = load <64 x i8>, <64 x i8>* %xptr
%y = load <64 x i8>, <64 x i8>* %yptr
%cmp = icmp ult <64 x i8> %x, %y
%max = select <64 x i1> %cmp, <64 x i8> %y, <64 x i8> %x
%res = sub <64 x i8> %max, %y
store <64 x i8> %res, <64 x i8>* %zptr
ret void
}

```

```

define
dso_local void @psubus_64i8_max_512(<64 x i8>* %xptr, <64 x i8>* %yptr, <64 x i8>* %zptr) "min-legal-
vector-width"="512" {
; CHECK-LABEL: psubus_64i8_max_512:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa64 (%rdi), %zmm0
; CHECK-NEXT: vpsubusb (%rsi), %zmm0, %zmm0
; CHECK-NEXT: vmovdqa64 %zmm0, (%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%x = load <64 x i8>, <64 x i8>* %xptr
%y = load <64 x i8>, <64 x i8>* %yptr
%cmp = icmp ult <64 x i8> %x, %y
%max = select <64 x i1> %cmp, <64 x i8> %y, <64 x i8> %x
%res = sub <64 x i8> %max, %y
store <64 x i8> %res, <64 x i8>* %zptr
ret void
}

```

```

define dso_local i32 @_Z9test_charPcS_i_256(i8* nocapture readonly, i8* nocapture readonly, i32) "min-legal-
vector-width"="256" {
; CHECK-SKX-LABEL: _Z9test_charPcS_i_256:
; CHECK-SKX:      # %bb.0: # %entry
; CHECK-SKX-NEXT: movl %edx, %eax
; CHECK-SKX-NEXT: vpxor %xmm0, %xmm0, %xmm0
; CHECK-SKX-NEXT: xorl %ecx, %ecx
; CHECK-SKX-NEXT: vpxor %xmm1, %xmm1, %xmm1
;
CHECK-SKX-NEXT: vpxor %xmm2, %xmm2, %xmm2
; CHECK-SKX-NEXT: .p2align 4, 0x90
; CHECK-SKX-NEXT: .LBB8_1: # %vector.body
; CHECK-SKX-NEXT: # =>This Inner Loop Header: Depth=1

```

```

; CHECK-SKX-NEXT: vpmovsxbw 16(%rdi,%rcx), %ymm3
; CHECK-SKX-NEXT: vpmovsxbw (%rdi,%rcx), %ymm4
; CHECK-SKX-NEXT: vpmovsxbw 16(%rsi,%rcx), %ymm5
; CHECK-SKX-NEXT: vpmaddwd %ymm3, %ymm5, %ymm3
; CHECK-SKX-NEXT: vpadd %ymm2, %ymm3, %ymm2
; CHECK-SKX-NEXT: vpmovsxbw (%rsi,%rcx), %ymm3
; CHECK-SKX-NEXT: vpmaddwd %ymm4, %ymm3, %ymm3
; CHECK-SKX-NEXT: vpadd %ymm1, %ymm3, %ymm1
; CHECK-SKX-NEXT: addq $32, %rcx
; CHECK-SKX-NEXT: cmpq %rcx, %rax
; CHECK-SKX-NEXT: jne .LBB8_1
; CHECK-SKX-NEXT: # %bb.2: # %middle.block
; CHECK-SKX-NEXT: vpadd %ymm0, %ymm1, %ymm1
; CHECK-SKX-NEXT: vpadd %ymm0, %ymm2, %ymm0
; CHECK-SKX-NEXT: vpadd %ymm0, %ymm1, %ymm0
; CHECK-SKX-NEXT: vextracti128 $1, %ymm0, %xmm1
; CHECK-SKX-NEXT: vpadd %xmm1, %xmm0, %xmm0
; CHECK-SKX-NEXT:
    vpsltd {{.*#+}} xmm1 = xmm0[2,3,2,3]
; CHECK-SKX-NEXT: vpadd %xmm1, %xmm0, %xmm0
; CHECK-SKX-NEXT: vpsrlq $32, %xmm0, %xmm1
; CHECK-SKX-NEXT: vpadd %xmm1, %xmm0, %xmm0
; CHECK-SKX-NEXT: vmovd %xmm0, %eax
; CHECK-SKX-NEXT: vzeroupper
; CHECK-SKX-NEXT: retq
;
; CHECK-AVX512-LABEL: _Z9test_charPcS_i_256:
; CHECK-AVX512:    # %bb.0: # %entry
; CHECK-AVX512-NEXT: movl %edx, %eax
; CHECK-AVX512-NEXT: vpxor %xmm0, %xmm0, %xmm0
; CHECK-AVX512-NEXT: xorl %ecx, %ecx
; CHECK-AVX512-NEXT: vpxor %xmm1, %xmm1, %xmm1
; CHECK-AVX512-NEXT: vpxor %xmm2, %xmm2, %xmm2
; CHECK-AVX512-NEXT: .p2align 4, 0x90
; CHECK-AVX512-NEXT: .LBB8_1: # %vector.body
; CHECK-AVX512-NEXT: # =>This Inner Loop Header: Depth=1
; CHECK-AVX512-NEXT: vpmovsxbw 16(%rdi,%rcx), %ymm3
; CHECK-AVX512-NEXT: vpmovsxbw (%rdi,%rcx), %ymm4
; CHECK-AVX512-NEXT: vpmovsxbw 16(%rsi,%rcx), %ymm5
; CHECK-AVX512-NEXT: vpmaddwd %ymm3, %ymm5, %ymm3
; CHECK-AVX512-NEXT: vpadd %ymm2, %ymm3,
    %ymm2
; CHECK-AVX512-NEXT: vpmovsxbw (%rsi,%rcx), %ymm3
; CHECK-AVX512-NEXT: vpmaddwd %ymm4, %ymm3, %ymm3
; CHECK-AVX512-NEXT: vpadd %ymm1, %ymm3, %ymm1
; CHECK-AVX512-NEXT: addq $32, %rcx
; CHECK-AVX512-NEXT: cmpq %rcx, %rax
; CHECK-AVX512-NEXT: jne .LBB8_1

```

```

; CHECK-AVX512-NEXT: # %bb.2: # %middle.block
; CHECK-AVX512-NEXT: vpaddd %ymm0, %ymm1, %ymm1
; CHECK-AVX512-NEXT: vpaddd %ymm0, %ymm2, %ymm0
; CHECK-AVX512-NEXT: vpaddd %ymm0, %ymm1, %ymm0
; CHECK-AVX512-NEXT: vextracti128 $1, %ymm0, %xmm1
; CHECK-AVX512-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-AVX512-NEXT: vpslufd {{.*#+}} xmm1 = xmm0[2,3,2,3]
; CHECK-AVX512-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-AVX512-NEXT: vpsrlq $32, %xmm0, %xmm1
; CHECK-AVX512-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-AVX512-NEXT: vmovd %xmm0, %eax
; CHECK-AVX512-NEXT: vzeroupper
; CHECK-AVX512-NEXT: retq
;
; CHECK-VBMI-LABEL: _Z9test_charPcS_i_256:
; CHECK-VBMI: # %bb.0: # %entry
; CHECK-VBMI-NEXT:
    movl %edx, %eax
; CHECK-VBMI-NEXT: vpxor %xmm0, %xmm0, %xmm0
; CHECK-VBMI-NEXT: xorl %ecx, %ecx
; CHECK-VBMI-NEXT: vpxor %xmm1, %xmm1, %xmm1
; CHECK-VBMI-NEXT: vpxor %xmm2, %xmm2, %xmm2
; CHECK-VBMI-NEXT: .p2align 4, 0x90
; CHECK-VBMI-NEXT: .LBB8_1: # %vector.body
; CHECK-VBMI-NEXT: # =>This Inner Loop Header: Depth=1
; CHECK-VBMI-NEXT: vpmovsxbw 16(%rdi,%rcx), %ymm3
; CHECK-VBMI-NEXT: vpmovsxbw (%rdi,%rcx), %ymm4
; CHECK-VBMI-NEXT: vpmovsxbw 16(%rsi,%rcx), %ymm5
; CHECK-VBMI-NEXT: vpmaddwd %ymm3, %ymm5, %ymm3
; CHECK-VBMI-NEXT: vpaddd %ymm2, %ymm3, %ymm2
; CHECK-VBMI-NEXT: vpmovsxbw (%rsi,%rcx), %ymm3
; CHECK-VBMI-NEXT: vpmaddwd %ymm4, %ymm3, %ymm3
; CHECK-VBMI-NEXT: vpaddd %ymm1, %ymm3, %ymm1
; CHECK-VBMI-NEXT: addq $32, %rcx
; CHECK-VBMI-NEXT: cmpq %rcx, %rax
; CHECK-VBMI-NEXT: jne .LBB8_1
; CHECK-VBMI-NEXT: # %bb.2: # %middle.block
; CHECK-VBMI-NEXT: vpaddd %ymm0, %ymm1, %ymm1
; CHECK-VBMI-NEXT: vpaddd %ymm0, %ymm2,
    %ymm0
; CHECK-VBMI-NEXT: vpaddd %ymm0, %ymm1, %ymm0
; CHECK-VBMI-NEXT: vextracti128 $1, %ymm0, %xmm1
; CHECK-VBMI-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-VBMI-NEXT: vpslufd {{.*#+}} xmm1 = xmm0[2,3,2,3]
; CHECK-VBMI-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-VBMI-NEXT: vpslufd {{.*#+}} xmm1 = xmm0[1,1,1,1]
; CHECK-VBMI-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-VBMI-NEXT: vmovd %xmm0, %eax

```

; CHECK-VBMI-NEXT: vzeroupper

; CHECK-VBMI-NEXT: retq

entry:

%3 = zext i32 %2 to i64

br label %vector.body

vector.body:

%index = phi i64 [ %index.next, %vector.body ], [ 0, %entry ]

%vec.phi = phi <32 x i32> [ %11, %vector.body ], [ zeroinitializer, %entry ]

%4 = getelementptr inbounds i8, i8\* %0, i64 %index

%5 = bitcast i8\* %4 to <32 x i8>\*

%wide.load = load <32 x i8>, <32 x i8>\* %5, align 1

%6 = sext <32 x i8> %wide.load to <32 x i32>

%7 = getelementptr inbounds i8, i8\* %1, i64 %index

%8 = bitcast i8\* %7 to <32 x i8>\*

%wide.load14 = load <32 x

i8>, <32 x i8>\* %8, align 1

%9 = sext <32 x i8> %wide.load14 to <32 x i32>

%10 = mul nsw <32 x i32> %9, %6

%11 = add nsw <32 x i32> %10, %vec.phi

%index.next = add i64 %index, 32

%12 = icmp eq i64 %index.next, %3

br i1 %12, label %middle.block, label %vector.body

middle.block:

%rdx.shuf1 = shufflevector <32 x i32> %11, <32 x i32> undef, <32 x i32> <i32 16, i32 17, i32 18, i32 19, i32 20, i32 21, i32 22, i32 23, i32 24, i32 25, i32 26, i32 27, i32 28, i32 29, i32 30, i32 31, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>

%bin.rdx1 = add <32 x i32> %11, %rdx.shuf1

%rdx.shuf = shufflevector <32 x i32> %bin.rdx1, <32 x i32> undef, <32 x i32> <i32 8, i32 9, i32 10, i32 11, i32 12, i32 13, i32 14, i32 15, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>

%bin.rdx = add <32 x i32> %bin.rdx1, %rdx.shuf

%rdx.shuf15 = shufflevector <32 x i32> %bin.rdx, <32 x i32> undef, <32 x i32> <i32 4, i32 5, i32 6, i32 7, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>

%bin.rdx32 = add <32 x i32> %bin.rdx, %rdx.shuf15

%rdx.shuf17 = shufflevector <32 x i32> %bin.rdx32, <32 x i32> undef, <32 x i32> <i32 2, i32 3, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>

%bin.rdx18 = add <32 x i32> %bin.rdx32, %rdx.shuf17

%rdx.shuf19 = shufflevector <32 x i32> %bin.rdx18, <32 x i32> undef, <32 x i32> <i32 1, i32 undef, i32 undef,

```

i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx20 = add <32 x i32> %bin.rdx18, %rdx.shuf19
%13 = extractelement <32 x i32> %bin.rdx20, i32 0
ret i32 %13
}

```

```

; CHECK-AVX512-NEXT: .p2align 4, 0x90
; CHECK-AVX512-NEXT: .LBB9_1: # %vector.body
; CHECK-AVX512-NEXT: # =>This Inner Loop Header: Depth=1
; CHECK-AVX512-NEXT: vpmovsxbw (%rdi,%rcx), %zmm2
; CHECK-AVX512-NEXT: vpmovsxbw (%rsi,%rcx), %zmm3
; CHECK-AVX512-NEXT: vpmaddwd %zmm2, %zmm3, %zmm2
; CHECK-AVX512-NEXT: vpaddd %zmm1, %zmm2, %zmm1
; CHECK-AVX512-NEXT: addq $32, %rcx
; CHECK-AVX512-NEXT: cmpq %rcx, %rax
; CHECK-AVX512-NEXT: jne .LBB9_1
; CHECK-AVX512-NEXT: # %bb.2: # %middle.block
; CHECK-AVX512-NEXT: vpaddd %zmm0, %zmm1, %zmm0
; CHECK-AVX512-NEXT: vextracti64x4 $1, %zmm0, %ymm1
; CHECK-AVX512-NEXT: vpaddd
%zmm1, %zmm0, %zmm0
; CHECK-AVX512-NEXT: vextracti128 $1, %ymm0, %xmm1
; CHECK-AVX512-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-AVX512-NEXT: vpslufd {{.*#+}} xmm1 = xmm0[2,3,2,3]
; CHECK-AVX512-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-AVX512-NEXT: vpsrlq $32, %xmm0, %xmm1
; CHECK-AVX512-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-AVX512-NEXT: vmovd %xmm0, %eax
; CHECK-AVX512-NEXT: vzeroupper
; CHECK-AVX512-NEXT: retq
;
; CHECK-VBMI-LABEL: _Z9test_charPcS_i_512:
; CHECK-VBMI: # %bb.0: # %entry
; CHECK-VBMI-NEXT: movl %edx, %eax
; CHECK-VBMI-NEXT: vpxor %xmm0, %xmm0, %xmm0
; CHECK-VBMI-NEXT: xorl %ecx, %ecx
; CHECK-VBMI-NEXT: vpxor %xmm1, %xmm1, %xmm1
; CHECK-VBMI-NEXT: .p2align 4, 0x90
; CHECK-VBMI-NEXT: .LBB9_1: # %vector.body
; CHECK-VBMI-NEXT: # =>This Inner Loop Header: Depth=1
; CHECK-VBMI-NEXT: vpmovsxbw (%rdi,%rcx), %zmm2
; CHECK-VBMI-NEXT: vpmovsxbw (%rsi,%rcx), %zmm3
; CHECK-VBMI-NEXT: vpmaddwd %zmm2, %zmm3, %zmm2
;
CHECK-VBMI-NEXT: vpaddd %zmm1, %zmm2, %zmm1
; CHECK-VBMI-NEXT: addq $32, %rcx
; CHECK-VBMI-NEXT: cmpq %rcx, %rax
; CHECK-VBMI-NEXT: jne .LBB9_1
; CHECK-VBMI-NEXT: # %bb.2: # %middle.block
; CHECK-VBMI-NEXT: vpaddd %zmm0, %zmm1, %zmm0
; CHECK-VBMI-NEXT: vextracti64x4 $1, %zmm0, %ymm1
; CHECK-VBMI-NEXT: vpaddd %zmm1, %zmm0, %zmm0
; CHECK-VBMI-NEXT: vextracti128 $1, %ymm0, %xmm1
; CHECK-VBMI-NEXT: vpaddd %xmm1, %xmm0, %xmm0

```



```
; CHECK-VBMI-NEXT:  vpsshufd {{.*#+}} xmm1 = xmm0[2,3,2,3]
; CHECK-VBMI-NEXT:  vpaddd %xmm1, %xmm0, %xmm0
; CHECK-VBMI-NEXT:  vpsshufd {{.*#+}} xmm1 = xmm0[1,1,1,1]
; CHECK-VBMI-NEXT:  vpaddd %xmm1, %xmm0, %xmm0
; CHECK-VBMI-NEXT:  vmovd %xmm0, %eax
; CHECK-VBMI-NEXT:  vzeroupper
; CHECK-VBMI-NEXT:  retq
```

entry:

```
%3 = zext i32 %2 to i64
```

```
br label %vector.body
```

vector.body:

```
%index = phi i64 [ %index.next, %vector.body ], [ 0, %entry ]
```

```
%vec.phi = phi <32 x i32> [ %11, %vector.body ], [ zeroinitializer, %entry ]
```

```
%4 = getelementptr
```

```
inbounds i8, i8* %0, i64 %index
```

```
%5 = bitcast i8* %4 to <32 x i8>*
```

```
%wide.load = load <32 x i8>, <32 x i8>* %5, align 1
```

```
%6 = sext <32 x i8> %wide.load to <32 x i32>
```

```
%7 = getelementptr inbounds i8, i8* %1, i64 %index
```

```
%8 = bitcast i8* %7 to <32 x i8>*
```

```
%wide.load14 = load <32 x i8>, <32 x i8>* %8, align 1
```

```
%9 = sext <32 x i8> %wide.load14 to <32 x i32>
```

```
%10 = mul nsw <32 x i32> %9, %6
```

```
%11 = add nsw <32 x i32> %10, %vec.phi
```

```
%index.next = add i64 %index, 32
```

```
%12 = icmp eq i64 %index.next, %3
```

```
br i1 %12, label %middle.block, label %vector.body
```

middle.block:

```
%rdx.shuf1 = shufflevector <32 x i32> %11, <32 x i32> undef, <32 x i32> <i32 16, i32 17, i32 18, i32 19, i32 20,
i32 21, i32 22, i32 23, i32 24, i32 25, i32 26, i32 27, i32 28, i32 29, i32 30, i32 31, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef>
```

```
%bin.rdx1 = add <32 x i32> %11, %rdx.shuf1
```

```
%rdx.shuf = shufflevector <32 x i32> %bin.rdx1, <32 x i32> undef, <32 x i32> <i32 8, i32 9, i32 10, i32 11, i32
12, i32 13, i32 14, i32 15, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef>
```

```
%bin.rdx = add <32 x i32> %bin.rdx1, %rdx.shuf
```

```
%rdx.shuf15 = shufflevector <32 x i32> %bin.rdx, <32 x i32> undef, <32 x i32> <i32 4, i32 5, i32 6, i32 7, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
```

```
%bin.rdx32 = add <32 x i32> %bin.rdx, %rdx.shuf15
```

```
%rdx.shuf17 = shufflevector <32 x i32> %bin.rdx32, <32 x i32> undef,
<32 x i32> <i32 2, i32 3, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef>
```

```

i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef>
%bin.rdx18 = add <32 x i32> %bin.rdx32, %rdx.shuf17
%rdx.shuf19 = shufflevector <32 x i32> %bin.rdx18, <32 x i32> undef, <32 x i32> <i32 1, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx20 = add <32 x i32> %bin.rdx18, %rdx.shuf19
%13 = extractelement <32 x i32> %bin.rdx20, i32 0
ret i32 %13
}

```

```

@a = dso_local global [1024 x i8]
zeroinitializer, align 16
@b = dso_local global [1024 x i8] zeroinitializer, align 16

```

```

define dso_local i32 @sad_16i8_256() "min-legal-vector-width"="256" {
; CHECK-SKX-LABEL: sad_16i8_256:
; CHECK-SKX:      # %bb.0: # %entry
; CHECK-SKX-NEXT: vpxor %xmm0, %xmm0, %xmm0
; CHECK-SKX-NEXT: movq $-1024, %rax # imm = 0xFC00
; CHECK-SKX-NEXT: vpxor %xmm1, %xmm1, %xmm1
; CHECK-SKX-NEXT: .p2align 4, 0x90
; CHECK-SKX-NEXT: .LBB10_1: # %vector.body
; CHECK-SKX-NEXT: # =>This Inner Loop Header: Depth=1
; CHECK-SKX-NEXT: vmovdqu a+1024(%rax), %xmm2
; CHECK-SKX-NEXT: vpsadbw b+1024(%rax), %xmm2, %xmm2
; CHECK-SKX-NEXT: vpaddq %ymm1, %ymm2, %ymm1
; CHECK-SKX-NEXT: addq $4, %rax
; CHECK-SKX-NEXT: jne .LBB10_1
; CHECK-SKX-NEXT: # %bb.2: # %middle.block
; CHECK-SKX-NEXT: vpaddq %ymm0, %ymm1, %ymm0
; CHECK-SKX-NEXT: vextracti128 $1, %ymm0, %xmm1
; CHECK-SKX-NEXT: vpaddq %xmm1, %xmm0, %xmm0
; CHECK-SKX-NEXT: vpsltd {{.*#+}} xmm1 = xmm0[2,3,2,3]
; CHECK-SKX-NEXT:
; CHECK-SKX-NEXT: vpaddq %xmm1, %xmm0, %xmm0
; CHECK-SKX-NEXT: vpsrlq $32, %xmm0, %xmm1
; CHECK-SKX-NEXT: vpaddq %xmm1, %xmm0, %xmm0
; CHECK-SKX-NEXT: vmovd %xmm0, %eax
; CHECK-SKX-NEXT: vzeroupper
; CHECK-SKX-NEXT: retq
;
; CHECK-AVX512-LABEL: sad_16i8_256:
; CHECK-AVX512:      # %bb.0: # %entry
; CHECK-AVX512-NEXT: vpxor %xmm0, %xmm0, %xmm0
; CHECK-AVX512-NEXT: movq $-1024, %rax # imm = 0xFC00
; CHECK-AVX512-NEXT: vpxor %xmm1, %xmm1, %xmm1

```

```

; CHECK-AVX512-NEXT: .p2align 4, 0x90
; CHECK-AVX512-NEXT: .LBB10_1: # %vector.body
; CHECK-AVX512-NEXT: # =>This Inner Loop Header: Depth=1
; CHECK-AVX512-NEXT: vmovdqu a+1024(%rax), %xmm2
; CHECK-AVX512-NEXT: vpsadbw b+1024(%rax), %xmm2, %xmm2
; CHECK-AVX512-NEXT: vpaddd %ymm1, %ymm2, %ymm1
; CHECK-AVX512-NEXT: addq $4, %rax
; CHECK-AVX512-NEXT: jne .LBB10_1
; CHECK-AVX512-NEXT: # %bb.2: # %middle.block
; CHECK-AVX512-NEXT: vpaddd %ymm0, %ymm1, %ymm0
; CHECK-AVX512-NEXT: vextracti128 $1, %ymm0, %xmm1
;
CHECK-AVX512-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-AVX512-NEXT: vpslufd {{.*#+}} xmm1 = xmm0[2,3,2,3]
; CHECK-AVX512-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-AVX512-NEXT: vpsrlq $32, %xmm0, %xmm1
; CHECK-AVX512-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-AVX512-NEXT: vmovd %xmm0, %eax
; CHECK-AVX512-NEXT: vzeroupper
; CHECK-AVX512-NEXT: retq
;
; CHECK-VBMI-LABEL: sad_16i8_256:
; CHECK-VBMI: # %bb.0: # %entry
; CHECK-VBMI-NEXT: vpxor %xmm0, %xmm0, %xmm0
; CHECK-VBMI-NEXT: movq $-1024, %rax # imm = 0xFC00
; CHECK-VBMI-NEXT: vpxor %xmm1, %xmm1, %xmm1
; CHECK-VBMI-NEXT: .p2align 4, 0x90
; CHECK-VBMI-NEXT: .LBB10_1: # %vector.body
; CHECK-VBMI-NEXT: # =>This Inner Loop Header: Depth=1
; CHECK-VBMI-NEXT: vmovdqu a+1024(%rax), %xmm2
; CHECK-VBMI-NEXT: vpsadbw b+1024(%rax), %xmm2, %xmm2
; CHECK-VBMI-NEXT: vpaddd %ymm1, %ymm2, %ymm1
; CHECK-VBMI-NEXT: addq $4, %rax
; CHECK-VBMI-NEXT: jne .LBB10_1
; CHECK-VBMI-NEXT: # %bb.2: # %middle.block
;
CHECK-VBMI-NEXT: vpaddd %ymm0, %ymm1, %ymm0
; CHECK-VBMI-NEXT: vextracti128 $1, %ymm0, %xmm1
; CHECK-VBMI-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-VBMI-NEXT: vpslufd {{.*#+}} xmm1 = xmm0[2,3,2,3]
; CHECK-VBMI-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-VBMI-NEXT: vpslufd {{.*#+}} xmm1 = xmm0[1,1,1,1]
; CHECK-VBMI-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-VBMI-NEXT: vmovd %xmm0, %eax
; CHECK-VBMI-NEXT: vzeroupper
; CHECK-VBMI-NEXT: retq
entry:
br label %vector.body

```

vector.body:

```
%index = phi i64 [ 0, %entry ], [ %index.next, %vector.body ]
%vec.phi = phi <16 x i32> [ zeroinitializer, %entry ], [ %10, %vector.body ]
%0 = getelementptr inbounds [1024 x i8], [1024 x i8]* @a, i64 0, i64 %index
%1 = bitcast i8* %0 to <16 x i8>*
%wide.load = load <16 x i8>, <16 x i8>* %1, align 4
%2 = zext <16 x i8> %wide.load to <16 x i32>
%3 = getelementptr inbounds [1024 x i8], [1024 x i8]* @b, i64 0, i64 %index
%4 = bitcast i8* %3 to <16 x i8>*
%wide.load1
= load <16 x i8>, <16 x i8>* %4, align 4
%5 = zext <16 x i8> %wide.load1 to <16 x i32>
%6 = sub nsw <16 x i32> %2, %5
%7 = icmp sgt <16 x i32> %6, <i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1>
%8 = sub nsw <16 x i32> zeroinitializer, %6
%9 = select <16 x i1> %7, <16 x i32> %6, <16 x i32> %8
%10 = add nsw <16 x i32> %9, %vec.phi
%index.next = add i64 %index, 4
%11 = icmp eq i64 %index.next, 1024
br i1 %11, label %middle.block, label %vector.body
```

middle.block:

```
%rdx.shuf = shufflevector <16 x i32> %10, <16 x i32> undef, <16 x i32> <i32 8, i32 9, i32 10, i32 11, i32 12, i32 13, i32 14, i32 15, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx = add <16 x i32> %10, %rdx.shuf
%rdx.shuf2 = shufflevector <16 x i32> %bin.rdx, <16 x i32> undef, <16 x i32> <i32 4, i32 5, i32 6, i32 7, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx2 = add <16 x i32> %bin.rdx, %rdx.shuf2
%rdx.shuf3 = shufflevector <16 x i32> %bin.rdx2, <16 x i32> undef, <16 x i32> <i32 2, i32 3, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx3 = add <16 x i32> %bin.rdx2, %rdx.shuf3
%rdx.shuf4 = shufflevector <16 x i32> %bin.rdx3, <16 x i32> undef, <16 x i32> <i32 1, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx4 = add <16 x i32> %bin.rdx3, %rdx.shuf4
%12 = extractelement <16 x i32> %bin.rdx4, i32 0
ret i32 %12
}
```

```
define dso_local i32 @sad_16i8_512() "min-legal-vector-width"="512" {
; CHECK-SKX-LABEL: sad_16i8_512:
; CHECK-SKX:      # %bb.0: # %entry
; CHECK-SKX-NEXT:  vpxor %xmm0, %xmm0, %xmm0
; CHECK-SKX-NEXT:
```

```

    movq $-1024, %rax # imm = 0xFC00
; CHECK-SKX-NEXT: .p2align 4, 0x90
; CHECK-SKX-NEXT: .LBB11_1: # %vector.body
; CHECK-SKX-NEXT: # =>This Inner Loop Header: Depth=1
; CHECK-SKX-NEXT: vmovdqu a+1024(%rax), %xmm1
; CHECK-SKX-NEXT: vpsadbw b+1024(%rax), %xmm1, %xmm1
; CHECK-SKX-NEXT: vpadd %zmm0, %zmm1, %zmm0
; CHECK-SKX-NEXT: addq $4, %rax
; CHECK-SKX-NEXT: jne .LBB11_1
; CHECK-SKX-NEXT: # %bb.2: # %middle.block
; CHECK-SKX-NEXT: vextracti64x4 $1, %zmm0, %ymm1
; CHECK-SKX-NEXT: vpadd %zmm1, %zmm0, %zmm0
; CHECK-SKX-NEXT: vextracti128 $1, %ymm0, %xmm1
; CHECK-SKX-NEXT: vpadd %xmm1, %xmm0, %xmm0
; CHECK-SKX-NEXT: vpslq $32, %xmm0, %xmm1
; CHECK-SKX-NEXT: vpadd %xmm1, %xmm0, %xmm0
; CHECK-SKX-NEXT: vmovd %xmm0, %eax
; CHECK-SKX-NEXT: vzeroupper
; CHECK-SKX-NEXT: retq
;
; CHECK-AVX512-LABEL: sad_16i8_512:
; CHECK-AVX512:
    # %bb.0: # %entry
; CHECK-AVX512-NEXT: vpxor %xmm0, %xmm0, %xmm0
; CHECK-AVX512-NEXT: movq $-1024, %rax # imm = 0xFC00
; CHECK-AVX512-NEXT: .p2align 4, 0x90
; CHECK-AVX512-NEXT: .LBB11_1: # %vector.body
; CHECK-AVX512-NEXT: # =>This Inner Loop Header: Depth=1
; CHECK-AVX512-NEXT: vmovdqu a+1024(%rax), %xmm1
; CHECK-AVX512-NEXT: vpsadbw b+1024(%rax), %xmm1, %xmm1
; CHECK-AVX512-NEXT: vpadd %zmm0, %zmm1, %zmm0
; CHECK-AVX512-NEXT: addq $4, %rax
; CHECK-AVX512-NEXT: jne .LBB11_1
; CHECK-AVX512-NEXT: # %bb.2: # %middle.block
; CHECK-AVX512-NEXT: vextracti64x4 $1, %zmm0, %ymm1
; CHECK-AVX512-NEXT: vpadd %zmm1, %zmm0, %zmm0
; CHECK-AVX512-NEXT: vextracti128 $1, %ymm0, %xmm1
; CHECK-AVX512-NEXT: vpadd %xmm1, %xmm0, %xmm0
; CHECK-AVX512-NEXT: vpslq $32, %xmm0, %xmm1
; CHECK-AVX512-NEXT: vpadd %xmm1, %xmm0, %xmm0
; CHECK-AVX512-NEXT: vmovd %xmm0, %eax
; CHECK-AVX512-NEXT: vzeroupper
; CHECK-AVX512-NEXT: retq

```

```

;
; CHECK-VBMI-LABEL: sad_16i8_512:
; CHECK-VBMI:      # %bb.0: # %entry
; CHECK-VBMI-NEXT: vpxor %xmm0, %xmm0, %xmm0
; CHECK-VBMI-NEXT: movq $-1024, %rax # imm = 0xFC00
; CHECK-VBMI-NEXT: .p2align 4, 0x90
; CHECK-VBMI-NEXT: .LBB11_1: # %vector.body
; CHECK-VBMI-NEXT: # =>This Inner Loop Header: Depth=1
; CHECK-VBMI-NEXT: vmovdqu a+1024(%rax), %xmm1
; CHECK-VBMI-NEXT: vpsadbw b+1024(%rax), %xmm1, %xmm1
; CHECK-VBMI-NEXT: vpadd %zmm0, %zmm1, %zmm0
; CHECK-VBMI-NEXT: addq $4, %rax
; CHECK-VBMI-NEXT: jne .LBB11_1
; CHECK-VBMI-NEXT: # %bb.2: # %middle.block
; CHECK-VBMI-NEXT: vextracti64x4 $1, %zmm0, %ymm1
; CHECK-VBMI-NEXT: vpadd %zmm1, %zmm0, %zmm0
; CHECK-VBMI-NEXT: vextracti128 $1, %ymm0, %xmm1
; CHECK-VBMI-NEXT: vpadd %xmm1, %xmm0, %xmm0
; CHECK-VBMI-NEXT: vpsltd {{.*#+}} xmm1 = xmm0[2,3,2,3]
; CHECK-VBMI-NEXT: vpadd %xmm1, %xmm0, %xmm0
; CHECK-VBMI-NEXT:
    vpsltd {{.*#+}} xmm1 = xmm0[1,1,1,1]
; CHECK-VBMI-NEXT: vpadd %xmm1, %xmm0, %xmm0
; CHECK-VBMI-NEXT: vmovd %xmm0, %eax
; CHECK-VBMI-NEXT: vzeroupper
; CHECK-VBMI-NEXT: retq
entry:
br label %vector.body

vector.body:
%index = phi i64 [ 0, %entry ], [ %index.next, %vector.body ]
%vec.phi = phi <16 x i32> [ zeroinitializer, %entry ], [ %10, %vector.body ]
%0 = getelementptr inbounds [1024 x i8], [1024 x i8]* @a, i64 0, i64 %index
%1 = bitcast i8* %0 to <16 x i8>*
%wide.load = load <16 x i8>, <16 x i8>* %1, align 4
%2 = zext <16 x i8> %wide.load to <16 x i32>
%3 = getelementptr inbounds [1024 x i8], [1024 x i8]* @b, i64 0, i64 %index
%4 = bitcast i8* %3 to <16 x i8>*
%wide.load1 = load <16 x i8>, <16 x i8>* %4, align 4
%5 = zext <16 x i8> %wide.load1 to <16 x i32>
%6 = sub nsw <16 x i32> %2, %5
%7 = icmp sgt <16 x i32> %6, <i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1>
-1, i32 -1, i32 -1, i32
-1, i32 -1>
%8 = sub nsw <16 x i32> zeroinitializer, %6
%9 = select <16 x i1> %7, <16 x i32> %6, <16 x i32> %8
%10 = add nsw <16 x i32> %9, %vec.phi
%index.next = add i64 %index, 4

```

```
%11 = icmp eq i64 %index.next, 1024
br i1 %11, label %middle.block, label %vector.body
```

```
middle.block:
```

```
%rdx.shuf = shufflevector <16 x i32> %10, <16 x i32> undef, <16 x i32> <i32 8, i32 9, i32 10, i32 11, i32 12, i32
13, i32 14, i32 15, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx = add <16 x i32> %10, %rdx.shuf
%rdx.shuf2 = shufflevector <16 x i32> %bin.rdx, <16 x i32> undef, <16 x i32> <i32 4, i32 5, i32 6, i32 7, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef>
%bin.rdx2 = add <16 x i32> %bin.rdx, %rdx.shuf2
%rdx.shuf3 = shufflevector <16 x i32> %bin.rdx2, <16 x i32> undef, <16 x i32> <i32 2, i32 3, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx3 = add <16 x i32> %bin.rdx2, %rdx.shuf3
%rdx.shuf4 = shufflevector <16 x i32> %bin.rdx3, <16 x i32> undef, <16 x i32> <i32 1, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef>
%bin.rdx4 = add <16 x i32> %bin.rdx3, %rdx.shuf4
%12 = extractelement <16 x i32> %bin.rdx4, i32 0
ret i32 %12
}
```

```
define dso_local void @sbto16f32_256(<16 x i16> %a, <16 x float>* %res) "min-legal-vector-width"="256" {
; CHECK-LABEL: sbto16f32_256:
; CHECK:      # %bb.0:
; CHECK-NEXT: vpmovw2m %ymm0, %k0
; CHECK-NEXT: kshiftrw $8, %k0, %k1
; CHECK-NEXT: vpmovm2d %k1, %ymm0
; CHECK-NEXT: vcvtdq2ps %ymm0, %ymm0
; CHECK-NEXT: vpmovm2d %k0, %ymm1
; CHECK-NEXT: vcvtdq2ps %ymm1, %ymm1
; CHECK-NEXT: vmovaps %ymm1, (%rdi)
; CHECK-NEXT: vmovaps %ymm0, 32(%rdi)
; CHECK-NEXT:
    vzeroupper
; CHECK-NEXT: retq
%mask = icmp slt <16 x i16> %a, zeroinitializer
%1 = sitofp <16 x i1> %mask to <16 x float>
store <16 x float> %1, <16 x float>* %res
ret void
}
```

```
define dso_local void @sbto16f32_512(<16 x i16> %a, <16 x float>* %res) "min-legal-vector-width"="512" {
; CHECK-LABEL: sbto16f32_512:
; CHECK:      # %bb.0:
; CHECK-NEXT: vpmovw2m %ymm0, %k0
; CHECK-NEXT: vpmovm2d %k0, %zmm0
```

```

; CHECK-NEXT:   vcvtdq2ps %zmm0, %zmm0
; CHECK-NEXT:   vmovaps %zmm0, (%rdi)
; CHECK-NEXT:   vzeroupper
; CHECK-NEXT:   retq
%mask = icmp slt <16 x i16> %a, zeroinitializer
%1 = sitofp <16 x i1> %mask to <16 x float>
store <16 x float> %1, <16 x float>* %res
ret void
}

```

```

define dso_local void @sbto16f64_256(<16 x i16> %a, <16 x double>* %res) "min-legal-vector-width"="256" {
; CHECK-LABEL: sbto16f64_256:
; CHECK:       # %bb.0:
; CHECK-NEXT:   vpmovw2m %ymm0, %k0
; CHECK-NEXT:   kshiftrw $8, %k0, %k1
; CHECK-NEXT:   vpmovm2d %k1, %ymm0
; CHECK-NEXT:
    vcvtdq2pd %xmm0, %ymm1
; CHECK-NEXT:   vextracti128 $1, %ymm0, %xmm0
; CHECK-NEXT:   vcvtdq2pd %xmm0, %ymm0
; CHECK-NEXT:   vpmovm2d %k0, %ymm2
; CHECK-NEXT:   vcvtdq2pd %xmm2, %ymm3
; CHECK-NEXT:   vextracti128 $1, %ymm2, %xmm2
; CHECK-NEXT:   vcvtdq2pd %xmm2, %ymm2
; CHECK-NEXT:   vmovaps %ymm2, 32(%rdi)
; CHECK-NEXT:   vmovaps %ymm3, (%rdi)
; CHECK-NEXT:   vmovaps %ymm0, 96(%rdi)
; CHECK-NEXT:   vmovaps %ymm1, 64(%rdi)
; CHECK-NEXT:   vzeroupper
; CHECK-NEXT:   retq
%mask = icmp slt <16 x i16> %a, zeroinitializer
%1 = sitofp <16 x i1> %mask to <16 x double>
store <16 x double> %1, <16 x double>* %res
ret void
}

```

```

define dso_local void @sbto16f64_512(<16 x i16> %a, <16 x double>* %res) "min-legal-vector-width"="512" {
; CHECK-LABEL: sbto16f64_512:
; CHECK:       # %bb.0:
; CHECK-NEXT:   vpmovw2m %ymm0, %k0
; CHECK-NEXT:   vpmovm2d %k0, %zmm0
; CHECK-NEXT:   vcvtdq2pd %ymm0, %zmm1
; CHECK-NEXT:   vextracti64x4 $1, %zmm0, %ymm0
; CHECK-NEXT:   vcvtdq2pd
    %ymm0, %zmm0
; CHECK-NEXT:   vmovaps %zmm0, 64(%rdi)
; CHECK-NEXT:   vmovaps %zmm1, (%rdi)
; CHECK-NEXT:   vzeroupper

```



```

; CHECK-NEXT:    retq
%mask = icmp slt <16 x i16> %a, zeroinitializer
%1 = sitofp <16 x i1> %mask to <16 x double>
store <16 x double> %1, <16 x double>* %res
ret void
}

define dso_local void @ubto16f32_256(<16 x i16> %a, <16 x float>* %res) "min-legal-vector-width"="256" {
; CHECK-LABEL: ubto16f32_256:
; CHECK:        # %bb.0:
; CHECK-NEXT:    vpmovw2m %ymm0, %k0
; CHECK-NEXT:    kshiftrw $8, %k0, %k1
; CHECK-NEXT:    vpmovm2d %k1, %ymm0
; CHECK-NEXT:    vpsrld $31, %ymm0, %ymm0
; CHECK-NEXT:    vcvt dq2ps %ymm0, %ymm0
; CHECK-NEXT:    vpmovm2d %k0, %ymm1
; CHECK-NEXT:    vpsrld $31, %ymm1, %ymm1
; CHECK-NEXT:    vcvt dq2ps %ymm1, %ymm1
; CHECK-NEXT:    vmovaps %ymm1, (%rdi)
; CHECK-NEXT:    vmovaps %ymm0, 32(%rdi)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
%mask = icmp slt <16 x i16> %a, zeroinitializer
%1 = uitofp <16 x i1> %mask to <16 x float>
store <16 x float> %1, <16 x float>* %res
ret void
}

define dso_local void @ubto16f32_512(<16 x i16> %a, <16 x float>* %res) "min-legal-vector-width"="512" {
; CHECK-LABEL: ubto16f32_512:
; CHECK:        # %bb.0:
; CHECK-NEXT:    vpmovw2m %ymm0, %k0
; CHECK-NEXT:    vpmovm2d %k0, %zmm0
; CHECK-NEXT:    vpsrld $31, %zmm0, %zmm0
; CHECK-NEXT:    vcvt dq2ps %zmm0, %zmm0
; CHECK-NEXT:    vmovaps %zmm0, (%rdi)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
%mask = icmp slt <16 x i16> %a, zeroinitializer
%1 = uitofp <16 x i1> %mask to <16 x float>
store <16 x float> %1, <16 x float>* %res
ret void
}

define dso_local void @ubto16f64_256(<16 x i16> %a, <16 x double>* %res) "min-legal-vector-width"="256" {
; CHECK-LABEL: ubto16f64_256:
; CHECK:        # %bb.0:
; CHECK-NEXT:    vpmovw2m %ymm0, %k0

```

```

; CHECK-NEXT: kshiftrw $8, %k0, %k1
; CHECK-NEXT: vpmovm2d %k1, %ymm0
; CHECK-NEXT: vpsrld $31, %ymm0, %ymm0
; CHECK-NEXT: vcvtdq2pd %xmm0, %ymm1
; CHECK-NEXT: vextracti128
$1, %ymm0, %xmm0
; CHECK-NEXT: vcvtdq2pd %xmm0, %ymm0
; CHECK-NEXT: vpmovm2d %k0, %ymm2
; CHECK-NEXT: vpsrld $31, %ymm2, %ymm2
; CHECK-NEXT: vcvtdq2pd %xmm2, %ymm3
; CHECK-NEXT: vextracti128 $1, %ymm2, %xmm2
; CHECK-NEXT: vcvtdq2pd %xmm2, %ymm2
; CHECK-NEXT: vmovaps %ymm2, 32(%rdi)
; CHECK-NEXT: vmovaps %ymm3, (%rdi)
; CHECK-NEXT: vmovaps %ymm0, 96(%rdi)
; CHECK-NEXT: vmovaps %ymm1, 64(%rdi)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%mask = icmp slt <16 x i16> %a, zeroinitializer
%1 = uitofp <16 x i1> %mask to <16 x double>
store <16 x double> %1, <16 x double>* %res
ret void
}

```

```

define dso_local void @ubto16f64_512(<16 x i16> %a, <16 x double>* %res) "min-legal-vector-width"="512" {
; CHECK-LABEL: ubto16f64_512:
; CHECK:      # %bb.0:
; CHECK-NEXT: vpmovw2m %ymm0, %k0
; CHECK-NEXT: vpmovm2d %k0, %zmm0
; CHECK-NEXT: vpsrld $31, %zmm0, %zmm0
; CHECK-NEXT: vcvtdq2pd %ymm0, %zmm1
; CHECK-NEXT: vextracti64x4 $1, %zmm0, %ymm0
; CHECK-NEXT:
    vcvtdq2pd %ymm0, %zmm0
; CHECK-NEXT: vmovaps %zmm0, 64(%rdi)
; CHECK-NEXT: vmovaps %zmm1, (%rdi)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%mask = icmp slt <16 x i16> %a, zeroinitializer
%1 = uitofp <16 x i1> %mask to <16 x double>
store <16 x double> %1, <16 x double>* %res
ret void
}

```

```

define <16 x i16> @test_16f32toub_256(<16 x float>* %ptr, <16 x i16> %passthru) "min-legal-vector-
width"="256" {
; CHECK-LABEL: test_16f32toub_256:
; CHECK:      # %bb.0:

```

```

; CHECK-NEXT: vcvttps2dq (%rdi), %ymm1
; CHECK-NEXT: vpslld $31, %ymm1, %ymm1
; CHECK-NEXT: vpmovd2m %ymm1, %k0
; CHECK-NEXT: vcvttps2dq 32(%rdi), %ymm1
; CHECK-NEXT: vpslld $31, %ymm1, %ymm1
; CHECK-NEXT: vpmovd2m %ymm1, %k1
; CHECK-NEXT: kunpckbw %k0, %k1, %k1
; CHECK-NEXT: vmovdqu16 %ymm0, %ymm0 {%k1} {z}
; CHECK-NEXT: retq
%a = load <16 x float>, <16 x float>* %ptr
%mask = fptoui <16 x float> %a to <16 x i1>
%select = select <16 x i1> %mask, <16 x i16> %passthru,
<16 x i16> zeroinitializer
ret <16 x i16> %select
}

```

```

define <16 x i16> @test_16f32toub_512(<16 x float>* %ptr, <16 x i16> %passthru) "min-legal-vector-
width"="512" {
; CHECK-LABEL: test_16f32toub_512:
; CHECK:      # %bb.0:
; CHECK-NEXT: vcvttps2dq (%rdi), %zmm1
; CHECK-NEXT: vpslld $31, %zmm1, %zmm1
; CHECK-NEXT: vpmovd2m %zmm1, %k1
; CHECK-NEXT: vmovdqu16 %ymm0, %ymm0 {%k1} {z}
; CHECK-NEXT: retq
%a = load <16 x float>, <16 x float>* %ptr
%mask = fptoui <16 x float> %a to <16 x i1>
%select = select <16 x i1> %mask, <16 x i16> %passthru, <16 x i16> zeroinitializer
ret <16 x i16> %select
}

```

```

define <16 x i16> @test_16f32tosb_256(<16 x float>* %ptr, <16 x i16> %passthru) "min-legal-vector-
width"="256" {
; CHECK-LABEL: test_16f32tosb_256:
; CHECK:      # %bb.0:
; CHECK-NEXT: vcvttps2dq (%rdi), %ymm1
; CHECK-NEXT: vpmovd2m %ymm1, %k0
; CHECK-NEXT: vcvttps2dq 32(%rdi), %ymm1
; CHECK-NEXT: vpmovd2m %ymm1, %k1
; CHECK-NEXT: kunpckbw %k0, %k1, %k1
; CHECK-NEXT:
    vmovdqu16 %ymm0, %ymm0 {%k1} {z}
; CHECK-NEXT: retq
%a = load <16 x float>, <16 x float>* %ptr
%mask = fptosi <16 x float> %a to <16 x i1>
%select = select <16 x i1> %mask, <16 x i16> %passthru, <16 x i16> zeroinitializer
ret <16 x i16> %select
}

```

```

define <16 x i16> @test_16f32tosb_512(<16 x float>* %ptr, <16 x i16> %passthru) "min-legal-vector-
width"="512" {
; CHECK-LABEL: test_16f32tosb_512:
; CHECK:      # %bb.0:
; CHECK-NEXT:  vcvttps2dq (%rdi), %zmm1
; CHECK-NEXT:  vpmovd2m %zmm1, %k1
; CHECK-NEXT:  vmovdqu16 %ymm0, %ymm0 {%k1} {z}
; CHECK-NEXT:  retq
%a = load <16 x float>, <16 x float>* %ptr
%mask = fptosi <16 x float> %a to <16 x i1>
%select = select <16 x i1> %mask, <16 x i16> %passthru, <16 x i16> zeroinitializer
ret <16 x i16> %select
}

```

```

define dso_local void @mul256(<64 x i8>* %a, <64 x i8>* %b, <64 x i8>* %c) "min-legal-vector-width"="256" {
; CHECK-SKX-VBMI-LABEL: mul256:
; CHECK-SKX-VBMI:      # %bb.0:
; CHECK-SKX-VBMI-NEXT:  vmovdqa
(%rdi), %ymm0
; CHECK-SKX-VBMI-NEXT:  vmovdqa 32(%rdi), %ymm1
; CHECK-SKX-VBMI-NEXT:  vmovdqa (%rsi), %ymm2
; CHECK-SKX-VBMI-NEXT:  vmovdqa 32(%rsi), %ymm3
; CHECK-SKX-VBMI-NEXT:  vpunpckhbw {{.*#+}} ymm4 =
ymm3[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-SKX-VBMI-NEXT:  vpunpckhbw {{.*#+}} ymm5 =
ymm1[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-SKX-VBMI-NEXT:  vpmullw %ymm4, %ymm5, %ymm4
; CHECK-SKX-VBMI-NEXT:  vpunpcklbw {{.*#+}} ymm3 =
ymm3[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-SKX-VBMI-NEXT:  vpunpcklbw {{.*#+}} ymm1 =
ymm1[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-SKX-VBMI-NEXT:  vpmullw %ymm3, %ymm1, %ymm1
; CHECK-SKX-VBMI-NEXT:  vmovdqa {{.*#+}} ymm3 =
[0,2,4,6,8,10,12,14,32,34,36,38,40,42,44,46,16,18,20,22,24,26,28,30,48,50,52,54,56,58,60,62]
;
CHECK-SKX-VBMI-NEXT:  vpermt2b %ymm4, %ymm3, %ymm1
; CHECK-SKX-VBMI-NEXT:  vpunpckhbw {{.*#+}} ymm4 =
ymm2[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-SKX-VBMI-NEXT:  vpunpckhbw {{.*#+}} ymm5 =
ymm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-SKX-VBMI-NEXT:  vpmullw %ymm4, %ymm5, %ymm4
; CHECK-SKX-VBMI-NEXT:  vpunpcklbw {{.*#+}} ymm2 =
ymm2[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-SKX-VBMI-NEXT:  vpunpcklbw {{.*#+}} ymm0 =
ymm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-SKX-VBMI-NEXT:  vpmullw %ymm2, %ymm0, %ymm0
; CHECK-SKX-VBMI-NEXT:  vpermt2b %ymm4, %ymm3, %ymm0

```

```

; CHECK-SKX-VBMI-NEXT: vmovdqa %ymm0, (%rdx)
; CHECK-SKX-VBMI-NEXT: vmovdqa %ymm1, 32(%rdx)
; CHECK-SKX-VBMI-NEXT: vzeroupper
; CHECK-SKX-VBMI-NEXT: retq
;
; CHECK-AVX512-LABEL: mul256:
; CHECK-AVX512:
    # %bb.0:
; CHECK-AVX512-NEXT: vmovdqa (%rdi), %ymm0
; CHECK-AVX512-NEXT: vmovdqa 32(%rdi), %ymm1
; CHECK-AVX512-NEXT: vmovdqa (%rsi), %ymm2
; CHECK-AVX512-NEXT: vmovdqa 32(%rsi), %ymm3
; CHECK-AVX512-NEXT: vpunpckhbw {{.*#+}} ymm4 =
ymm3[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-AVX512-NEXT: vpunpckhbw {{.*#+}} ymm5 =
ymm1[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-AVX512-NEXT: vpmullw %ymm4, %ymm5, %ymm4
; CHECK-AVX512-NEXT: vpbroadcastw {{.*#+}} ymm5 =
[255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255]
; CHECK-AVX512-NEXT: vpand %ymm5, %ymm4, %ymm4
; CHECK-AVX512-NEXT: vpunpcklbw {{.*#+}} ymm3 =
ymm3[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-AVX512-NEXT: vpunpcklbw {{.*#+}} ymm1 =
ymm1[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
;
CHECK-AVX512-NEXT: vpmullw %ymm3, %ymm1, %ymm1
; CHECK-AVX512-NEXT: vpand %ymm5, %ymm1, %ymm1
; CHECK-AVX512-NEXT: vpackuswb %ymm4, %ymm1, %ymm1
; CHECK-AVX512-NEXT: vpunpckhbw {{.*#+}} ymm3 =
ymm2[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-AVX512-NEXT: vpunpckhbw {{.*#+}} ymm4 =
ymm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-AVX512-NEXT: vpmullw %ymm3, %ymm4, %ymm3
; CHECK-AVX512-NEXT: vpand %ymm5, %ymm3, %ymm3
; CHECK-AVX512-NEXT: vpunpcklbw {{.*#+}} ymm2 =
ymm2[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-AVX512-NEXT: vpunpcklbw {{.*#+}} ymm0 =
ymm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-AVX512-NEXT: vpmullw %ymm2, %ymm0, %ymm0
; CHECK-AVX512-NEXT: vpand %ymm5, %ymm0, %ymm0
; CHECK-AVX512-NEXT: vpackuswb %ymm3, %ymm0, %ymm0
; CHECK-AVX512-NEXT:
    vmovdqa %ymm0, (%rdx)
; CHECK-AVX512-NEXT: vmovdqa %ymm1, 32(%rdx)
; CHECK-AVX512-NEXT: vzeroupper
; CHECK-AVX512-NEXT: retq
;
; CHECK-VBMI-LABEL: mul256:

```

```

; CHECK-VBMI:    # %bb.0:
; CHECK-VBMI-NEXT:  vmovdqa (%rdi), %ymm0
; CHECK-VBMI-NEXT:  vmovdqa 32(%rdi), %ymm1
; CHECK-VBMI-NEXT:  vmovdqa (%rsi), %ymm2
; CHECK-VBMI-NEXT:  vmovdqa 32(%rsi), %ymm3
; CHECK-VBMI-NEXT:  vpunpckhbw {{.*#+}} %ymm4 =
ymm3[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-VBMI-NEXT:  vpunpckhbw {{.*#+}} %ymm5 =
ymm1[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-VBMI-NEXT:  vpmullw %ymm4, %ymm5, %ymm4
; CHECK-VBMI-NEXT:  vpunpcklbw {{.*#+}} %ymm3 =
ymm3[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-VBMI-NEXT:  vpunpcklbw {{.*#+}} %ymm1 =
ymm1[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
;
CHECK-VBMI-NEXT:  vpmullw %ymm3, %ymm1, %ymm1
; CHECK-VBMI-NEXT:  vmovdqa {{.*#+}} %ymm3 =
[0,2,4,6,8,10,12,14,32,34,36,38,40,42,44,46,16,18,20,22,24,26,28,30,48,50,52,54,56,58,60,62]
; CHECK-VBMI-NEXT:  vpermt2b %ymm4, %ymm3, %ymm1
; CHECK-VBMI-NEXT:  vpunpckhbw {{.*#+}} %ymm4 =
ymm2[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-VBMI-NEXT:  vpunpckhbw {{.*#+}} %ymm5 =
ymm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-VBMI-NEXT:  vpmullw %ymm4, %ymm5, %ymm4
; CHECK-VBMI-NEXT:  vpunpcklbw {{.*#+}} %ymm2 =
ymm2[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-VBMI-NEXT:  vpunpcklbw {{.*#+}} %ymm0 =
ymm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-VBMI-NEXT:  vpmullw %ymm2, %ymm0, %ymm0
; CHECK-VBMI-NEXT:  vpermt2b %ymm4, %ymm3, %ymm0
; CHECK-VBMI-NEXT:  vmovdqa %ymm0, (%rdx)
;
CHECK-VBMI-NEXT:  vmovdqa %ymm1, 32(%rdx)
; CHECK-VBMI-NEXT:  vzeroupper
; CHECK-VBMI-NEXT:  retq
%d = load <64 x i8>, <64 x i8>* %a
%e = load <64 x i8>, <64 x i8>* %b
%f = mul <64 x i8> %d, %e
store <64 x i8> %f, <64 x i8>* %c
ret void
}

```

```

define dso_local void @mul512(<64 x i8>* %a, <64 x i8>* %b, <64 x i8>* %c) "min-legal-vector-width"="512" {

```

```

; CHECK-SKX-VBMI-LABEL: mul512:
; CHECK-SKX-VBMI:    # %bb.0:
; CHECK-SKX-VBMI-NEXT:  vmovdqa64 (%rdi), %zmm0
; CHECK-SKX-VBMI-NEXT:  vmovdqa64 (%rsi), %zmm1
; CHECK-SKX-VBMI-NEXT:  vpunpckhbw {{.*#+}} %zmm2 =

```

```

zmm1[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31,40,40,41,41,4
2,42,43,43,44,44,45,45,46,46,47,47,56,56,57,57,58,58,59,59,60,60,61,61,62,62,63,63]
; CHECK-SKX-VBMI-NEXT:  vpunpckhbw {{.*#+}} zmm3 =
zmm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31,40,40,41,41,4
2,42,43,43,44,44,45,45,46,46,47,47,56,56,57,57,58,58,59,59,60,60,61,61,62,62,63,63]
;
CHECK-SKX-VBMI-NEXT:  vmullw %zmm2, %zmm3, %zmm2
; CHECK-SKX-VBMI-NEXT:  vpunpcklbw {{.*#+}} zmm1 =
zmm1[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23,32,32,33,33,34,34,35,35,36,3
6,37,37,38,38,39,39,48,48,49,49,50,50,51,51,52,52,53,53,54,54,55,55]
; CHECK-SKX-VBMI-NEXT:  vpunpcklbw {{.*#+}} zmm0 =
zmm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23,32,32,33,33,34,34,35,35,36,3
6,37,37,38,38,39,39,48,48,49,49,50,50,51,51,52,52,53,53,54,54,55,55]
; CHECK-SKX-VBMI-NEXT:  vmullw %zmm1, %zmm0, %zmm0
; CHECK-SKX-VBMI-NEXT:  vmovdqa64 {{.*#+}} zmm1 =
[0,2,4,6,8,10,12,14,64,66,68,70,72,74,76,78,16,18,20,22,24,26,28,30,80,82,84,86,88,90,92,94,32,34,36,38,40,42,44,
46,96,98,100,102,104,106,108,110,48,50,52,54,56,58,60,62,112,114,116,118,120,122,124,126]
; CHECK-SKX-VBMI-NEXT:  vperm2b %zmm2, %zmm0, %zmm1
; CHECK-SKX-VBMI-NEXT:  vmovdqa64 %zmm1, (%rdx)
; CHECK-SKX-VBMI-NEXT:  vzeroupper
;
CHECK-SKX-VBMI-NEXT:  retq
;
; CHECK-AVX512-LABEL: mul512:
; CHECK-AVX512:  # %bb.0:
; CHECK-AVX512-NEXT:  vmovdqa64 (%rdi), %zmm0
; CHECK-AVX512-NEXT:  vmovdqa64 (%rsi), %zmm1
; CHECK-AVX512-NEXT:  vpunpckhbw {{.*#+}} zmm2 =
zmm1[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31,40,40,41,41,4
2,42,43,43,44,44,45,45,46,46,47,47,56,56,57,57,58,58,59,59,60,60,61,61,62,62,63,63]
; CHECK-AVX512-NEXT:  vpunpckhbw {{.*#+}} zmm3 =
zmm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31,40,40,41,41,4
2,42,43,43,44,44,45,45,46,46,47,47,56,56,57,57,58,58,59,59,60,60,61,61,62,62,63,63]
; CHECK-AVX512-NEXT:  vmullw %zmm2, %zmm3, %zmm2
; CHECK-AVX512-NEXT:  vpbroadcastw {{.*#+}} zmm3 =
[255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,255,255]
; CHECK-AVX512-NEXT:  vpandq %zmm3, %zmm2, %zmm2
; CHECK-AVX512-NEXT:  vpunpcklbw
{{.*#+}} zmm1 =
zmm1[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23,32,32,33,33,34,34,35,35,36,3
6,37,37,38,38,39,39,48,48,49,49,50,50,51,51,52,52,53,53,54,54,55,55]
; CHECK-AVX512-NEXT:  vpunpcklbw {{.*#+}} zmm0 =
zmm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23,32,32,33,33,34,34,35,35,36,3
6,37,37,38,38,39,39,48,48,49,49,50,50,51,51,52,52,53,53,54,54,55,55]
; CHECK-AVX512-NEXT:  vmullw %zmm1, %zmm0, %zmm0
; CHECK-AVX512-NEXT:  vpandq %zmm3, %zmm0, %zmm0
; CHECK-AVX512-NEXT:  vpackuswb %zmm2, %zmm0, %zmm0

```

```

; CHECK-AVX512-NEXT: vmovdqa64 %zmm0, (%rdx)
; CHECK-AVX512-NEXT: vzeroupper
; CHECK-AVX512-NEXT: retq
;
; CHECK-VBMI-LABEL: mul512:
; CHECK-VBMI:    # %bb.0:
; CHECK-VBMI-NEXT: vmovdqa64 (%rdi), %zmm0
; CHECK-VBMI-NEXT: vmovdqa64 (%rsi), %zmm1
; CHECK-VBMI-NEXT: vpunpckhbw {{.*#+}} %zmm2 =
zmm1[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31,40,40,41,41,4
2,42,43,43,44,44,45,45,46,46,47,47,56,56,57,57,58,58,59,59,60,60,61,61,62,62,63,63]
;
CHECK-VBMI-NEXT: vpunpckhbw {{.*#+}} %zmm3 =
zmm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31,40,40,41,41,4
2,42,43,43,44,44,45,45,46,46,47,47,56,56,57,57,58,58,59,59,60,60,61,61,62,62,63,63]
; CHECK-VBMI-NEXT: vpmullw %zmm2, %zmm3, %zmm2
; CHECK-VBMI-NEXT: vpunpcklbw {{.*#+}} %zmm1 =
zmm1[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23,32,32,33,33,34,34,35,35,36,3
6,37,37,38,38,39,39,48,48,49,49,50,50,51,51,52,52,53,53,54,54,55,55]
; CHECK-VBMI-NEXT: vpunpcklbw {{.*#+}} %zmm0 =
zmm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23,32,32,33,33,34,34,35,35,36,3
6,37,37,38,38,39,39,48,48,49,49,50,50,51,51,52,52,53,53,54,54,55,55]
; CHECK-VBMI-NEXT: vpmullw %zmm1, %zmm0, %zmm0
; CHECK-VBMI-NEXT: vmovdqa64 {{.*#+}} %zmm1 =
[0,2,4,6,8,10,12,14,64,66,68,70,72,74,76,78,16,18,20,22,24,26,28,30,80,82,84,86,88,90,92,94,32,34,36,38,40,42,44,
46,96,98,100,102,104,106,108,110,48,50,52,54,56,58,60,62,112,114,116,118,120,122,124,126]
;
CHECK-VBMI-NEXT: vperm2b %zmm2, %zmm0, %zmm1
; CHECK-VBMI-NEXT: vmovdqa64 %zmm1, (%rdx)
; CHECK-VBMI-NEXT: vzeroupper
; CHECK-VBMI-NEXT: retq
%d = load <64 x i8>, <64 x i8>* %a
%e = load <64 x i8>, <64 x i8>* %b
%f = mul <64 x i8> %d, %e
store <64 x i8> %f, <64 x i8>* %c
ret void
}

```

; This threw an assertion at one point.

```
define <4 x i32> @mload_v4i32(<4 x i32> %trigger, <4 x i32>* %addr, <4 x i32> %dst) "min-legal-vector-
width"="256" {
```

```

; CHECK-LABEL: mload_v4i32:
; CHECK:    # %bb.0:
; CHECK-NEXT: vptestnmd %xmm0, %xmm0, %k1
; CHECK-NEXT: vpbblendmd (%rdi), %xmm1, %xmm0 {%k1}
; CHECK-NEXT: retq
%mask = icmp eq <4 x i32> %trigger, zeroinitializer
%res = call <4 x i32> @llvm.masked.load.v4i32.p0v4i32(<4 x i32>* %addr, i32 4, <4 x i1> %mask, <4 x i32>

```



```

%dst)
ret <4 x i32> %res
}
declare <4 x i32>
@llvm.masked.load.v4i32.p0v4i32(<4 x i32>*, i32, <4 x i1>, <4 x i32>)

define <16 x i32> @trunc_v16i64_v16i32(<16 x i64>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_v16i64_v16i32:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rdi), %ymm0
; CHECK-NEXT: vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT: vmovdqa 64(%rdi), %ymm2
; CHECK-NEXT: vmovdqa 96(%rdi), %ymm3
; CHECK-NEXT: vpmovqd %ymm0, %xmm0
; CHECK-NEXT: vpmovqd %ymm1, %xmm1
; CHECK-NEXT: vinserti128 $1, %xmm1, %ymm0, %ymm0
; CHECK-NEXT: vpmovqd %ymm2, %xmm1
; CHECK-NEXT: vpmovqd %ymm3, %xmm2
; CHECK-NEXT: vinserti128 $1, %xmm2, %ymm1, %ymm1
; CHECK-NEXT: retq
;a = load <16 x i64>, <16 x i64>* %x
;b = trunc <16 x i64> %a to <16 x i32>
ret <16 x i32> %b
}

define <16 x i8> @trunc_v16i64_v16i8(<16 x i64>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_v16i64_v16i8:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rdi), %ymm0
; CHECK-NEXT: vmovdqa 32(%rdi),
;               %ymm1
; CHECK-NEXT: vmovdqa 64(%rdi), %ymm2
; CHECK-NEXT: vmovdqa 96(%rdi), %ymm3
; CHECK-NEXT: vpmovqb %ymm3, %xmm3
; CHECK-NEXT: vpmovqb %ymm2, %xmm2
; CHECK-NEXT: vpunpckldq {{.*#+}} xmm2 = xmm2[0],xmm3[0],xmm2[1],xmm3[1]
; CHECK-NEXT: vpmovqb %ymm1, %xmm1
; CHECK-NEXT: vpmovqb %ymm0, %xmm0
; CHECK-NEXT: vpunpckldq {{.*#+}} xmm0 = xmm0[0],xmm1[0],xmm0[1],xmm1[1]
; CHECK-NEXT: vpunpckldq {{.*#+}} xmm0 = xmm0[0],xmm2[0]
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
;a = load <16 x i64>, <16 x i64>* %x
;b = trunc <16 x i64> %a to <16 x i8>
ret <16 x i8> %b
}

define <16 x i8> @trunc_v16i32_v16i8(<16 x i32>* %x) nounwind "min-legal-vector-width"="256" {

```

```

; CHECK-LABEL: trunc_v16i32_v16i8:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rdi), %ymm0
; CHECK-NEXT: vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT: vpmovdb %ymm1, %xmm1
; CHECK-NEXT: vpmovdb %ymm0, %xmm0
; CHECK-NEXT: vpunpcklqdq {{.*#+}} xmm0 = xmm0[0],xmm1[0]
; CHECK-NEXT: vzeroupper
;
CHECK-NEXT:  retq
%a = load <16 x i32>, <16 x i32>* %x
%b = trunc <16 x i32> %a to <16 x i8>
ret <16 x i8> %b
}

define <8 x i8> @trunc_v8i64_v8i8(<8 x i64>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_v8i64_v8i8:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rdi), %ymm0
; CHECK-NEXT: vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT: vpmovqb %ymm1, %xmm1
; CHECK-NEXT: vpmovqb %ymm0, %xmm0
; CHECK-NEXT: vpunpckldq {{.*#+}} xmm0 = xmm0[0],xmm1[0],xmm0[1],xmm1[1]
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%a = load <8 x i64>, <8 x i64>* %x
%b = trunc <8 x i64> %a to <8 x i8>
ret <8 x i8> %b
}

define <8 x i16> @trunc_v8i64_v8i16(<8 x i64>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_v8i64_v8i16:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rdi), %ymm0
; CHECK-NEXT: vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT: vpmovqw %ymm1, %xmm1
; CHECK-NEXT: vpmovqw %ymm0, %xmm0
; CHECK-NEXT: vpunpcklqdq {{.*#+}} xmm0
= xmm0[0],xmm1[0]
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%a = load <8 x i64>, <8 x i64>* %x
%b = trunc <8 x i64> %a to <8 x i16>
ret <8 x i16> %b
}

define <8 x i32> @trunc_v8i64_v8i32_zeroes(<8 x i64>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_v8i64_v8i32_zeroes:

```

```

; CHECK:    # %bb.0:
; CHECK-NEXT:  vpsrlq $48, 32(%rdi), %ymm0
; CHECK-NEXT:  vpsrlq $48, (%rdi), %ymm1
; CHECK-NEXT:  vpackusdw %ymm0, %ymm1, %ymm0
; CHECK-NEXT:  vpermq {{.*#+}} ymm0 = ymm0[0,2,1,3]
; CHECK-NEXT:  retq
%a = load <8 x i64>, <8 x i64>* %x
%b = lshr <8 x i64> %a, <i64 48, i64 48, i64 48, i64 48, i64 48, i64 48, i64 48, i64 48>
%c = trunc <8 x i64> %b to <8 x i32>
ret <8 x i32> %c
}

define <16 x i16> @trunc_v16i32_v16i16_zeroes(<16 x i32>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_v16i32_v16i16_zeroes:
; CHECK:    # %bb.0:
; CHECK-NEXT:  vmovdqa (%rdi), %ymm1
; CHECK-NEXT:  vmovdqa {{.*#+}} ymm0 = [1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31]
;
; CHECK-NEXT:  vpermi2w 32(%rdi), %ymm1, %ymm0
; CHECK-NEXT:  retq
%a = load <16 x i32>, <16 x i32>* %x
%b = lshr <16 x i32> %a, <i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16>
%c = trunc <16 x i32> %b to <16 x i16>
ret <16 x i16> %c
}

define <32 x i8> @trunc_v32i16_v32i8_zeroes(<32 x i16>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-SKX-VBMI-LABEL: trunc_v32i16_v32i8_zeroes:
; CHECK-SKX-VBMI:    # %bb.0:
; CHECK-SKX-VBMI-NEXT:  vmovdqa (%rdi), %ymm1
; CHECK-SKX-VBMI-NEXT:  vmovdqa {{.*#+}} ymm0 =
[1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35,37,39,41,43,45,47,49,51,53,55,57,59,61,63]
; CHECK-SKX-VBMI-NEXT:  vpermi2b 32(%rdi), %ymm1, %ymm0
; CHECK-SKX-VBMI-NEXT:  retq
;
; CHECK-AVX512-LABEL: trunc_v32i16_v32i8_zeroes:
; CHECK-AVX512:    # %bb.0:
; CHECK-AVX512-NEXT:  vpsrlw $8, 32(%rdi), %ymm0
; CHECK-AVX512-NEXT:  vpsrlw $8, (%rdi),
%ymm1
; CHECK-AVX512-NEXT:  vpackuswb %ymm0, %ymm1, %ymm0
; CHECK-AVX512-NEXT:  vpermq {{.*#+}} ymm0 = ymm0[0,2,1,3]
; CHECK-AVX512-NEXT:  retq
;
; CHECK-VBMI-LABEL: trunc_v32i16_v32i8_zeroes:
; CHECK-VBMI:    # %bb.0:
; CHECK-VBMI-NEXT:  vmovdqa (%rdi), %ymm1

```

```
; CHECK-VBMI-NEXT:    vmovdqa {{.*#+}} ymm0 =
[1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35,37,39,41,43,45,47,49,51,53,55,57,59,61,63]
; CHECK-VBMI-NEXT:    vperm2b 32(%rdi), %ymm1, %ymm0
; CHECK-VBMI-NEXT:    retq

%a = load <32 x i16>, <32 x i16>* %x
%b = lshr <32 x i16> %a, <i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8>
%c = trunc <32 x i16> %b to <32 x i8>
ret <32 x i8> %c
}
```

```
define <16 x i16> @trunc_v16i32_v16i16_sign(<16 x i32>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_v16i32_v16i16_sign:
; CHECK:      # %bb.0:
; CHECK-NEXT:  vmovdqa (%rdi), %ymm1
; CHECK-NEXT:  vmovdqa {{.*#+}} ymm0 = [1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31]
; CHECK-NEXT:  vpermi2w 32(%rdi), %ymm1, %ymm0
; CHECK-NEXT:  retq
%a = load <16 x i32>, <16 x i32>* %x
%b = ashr <16 x i32> %a, <i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16>
%c = trunc <16 x i32> %b to <16
x i16>
ret <16 x i16> %c
}
```

```

; CHECK-SKX-VBMI-NEXT:  retq
;
; CHECK-AVX512-LABEL: trunc_v32i16_v32i8_sign:
; CHECK-AVX512:      # %bb.0:
; CHECK-AVX512-NEXT:  vpsrlw $8, 32(%rdi), %ymm0
; CHECK-AVX512-NEXT:  vpsrlw $8, (%rdi), %ymm1
; CHECK-AVX512-NEXT:  vpackuswb %ymm0, %ymm1, %ymm0
; CHECK-AVX512-NEXT:  vpermq {{.*#+}} ymm0 = ymm0[0,2,1,3]
; CHECK-AVX512-NEXT:  retq
;
; CHECK-VBMI-LABEL: trunc_v32i16_v32i8_sign:
; CHECK-VBMI:      # %bb.0:
; CHECK-VBMI-NEXT:  vmovdqa (%rdi), %ymm1
; CHECK-VBMI-NEXT:  vmovdqa {{.*#+}} ymm0 =
[1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35,37,39,41,43,45,47,49,51,53,55,57,59,61,63]
;
CHECK-VBMI-NEXT:  vpermi2b 32(%rdi), %ymm1, %ymm0
; CHECK-VBMI-NEXT:  retq
%a = load <32 x i16>, <32 x i16>* %x
%b = ashr <32 x i16> %a, <i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8>
%c = trunc <32 x i16> %b to <32 x i8>
ret <32 x i8> %c
}

```

```

define dso_local void @zext_v16i8_v16i64(<16 x i8> %x, <16 x i64>* %y) nounwind "min-legal-vector-
width"="256" {
; CHECK-LABEL: zext_v16i8_v16i64:
; CHECK:      # %bb.0:
; CHECK-NEXT:  vpmovxbw {{.*#+}} ymm1 =
xmm0[0],zero,xmm0[1],zero,xmm0[2],zero,xmm0[3],zero,xmm0[4],zero,xmm0[5],zero,xmm0[6],zero,xmm0[7],zer
o,xmm0[8],zero,xmm0[9],zero,xmm0[10],zero,xmm0[11],zero,xmm0[12],zero,xmm0[13],zero,xmm0[14],zero,xm
m0[15],zero
; CHECK-NEXT:  vpshufd {{.*#+}} xmm2 = xmm1[2,3,2,3]
; CHECK-NEXT:
vpmovzxwq {{.*#+}} ymm2 =
xmm2[0],zero,zero,zero,xmm2[1],zero,zero,zero,xmm2[2],zero,zero,zero,xmm2[3],zero,zero,zero
; CHECK-NEXT:  vextracti128 $1, %ymm1, %xmm1
; CHECK-NEXT:  vpshufd {{.*#+}} xmm3 = xmm1[2,3,2,3]
; CHECK-NEXT:  vpmovzxwq {{.*#+}} ymm3 =
xmm3[0],zero,zero,zero,xmm3[1],zero,zero,zero,xmm3[2],zero,zero,zero,xmm3[3],zero,zero,zero
; CHECK-NEXT:  vpmovzxwq {{.*#+}} ymm1 =
xmm1[0],zero,zero,zero,xmm1[1],zero,zero,zero,xmm1[2],zero,zero,zero,xmm1[3],zero,zero,zero
; CHECK-NEXT:  vpmovxbq {{.*#+}} ymm0 =
xmm0[0],zero,zero,zero,zero,zero,zero,zero,xmm0[1],zero,zero,zero,zero,zero,zero,zero,xmm0[2],zero,zero,zero,zer
o,zero,zero,zero,xmm0[3],zero,zero,zero,zero,zero,zero,zero,zero,zero,zero
; CHECK-NEXT:  vmovdqa %ymm0, (%rdi)
; CHECK-NEXT:  vmovdqa %ymm1, 64(%rdi)

```

```

; CHECK-NEXT: vmovdqa %ymm3, 96(%rdi)
; CHECK-NEXT: vmovdqa %ymm2, 32(%rdi)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%a = zext <16 x i8> %x to <16 x i64>
store <16 x i64> %a, <16 x i64>*
%y
ret void
}

```

```

define dso_local void @sext_v16i8_v16i64(<16 x i8> %x, <16 x i64>* %y) nounwind "min-legal-vector-
width"="256" {

```

```

; CHECK-LABEL: sext_v16i8_v16i64:
; CHECK:      # %bb.0:
; CHECK-NEXT: vpmovsxbw %xmm0, %ymm1
; CHECK-NEXT: vpshufd {{.*#+}} xmm2 = xmm1[2,3,2,3]
; CHECK-NEXT: vpmovsxdq %xmm2, %ymm2
; CHECK-NEXT: vextracti128 $1, %ymm1, %xmm1
; CHECK-NEXT: vpshufd {{.*#+}} xmm3 = xmm1[2,3,2,3]
; CHECK-NEXT: vpmovsxdq %xmm3, %ymm3
; CHECK-NEXT: vpmovsxdq %xmm1, %ymm1
; CHECK-NEXT: vpmovsxbq %xmm0, %ymm0
; CHECK-NEXT: vmovdqa %ymm0, (%rdi)
; CHECK-NEXT: vmovdqa %ymm1, 64(%rdi)
; CHECK-NEXT: vmovdqa %ymm3, 96(%rdi)
; CHECK-NEXT: vmovdqa %ymm2, 32(%rdi)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%a = sext <16 x i8> %x to <16 x i64>
store <16 x i64> %a, <16 x i64>* %y
ret void
}

```

```

define dso_local void @vselect_split_v8i16_setcc(<8 x i16> %s, <8 x i16> %t, <8 x i64>* %p, <8 x i64>* %q, <8
x i64>* %r) "min-legal-vector-width"="256"
{

```

```

; CHECK-LABEL: vselect_split_v8i16_setcc:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rsi), %ymm2
; CHECK-NEXT: vmovdqa 32(%rsi), %ymm3
; CHECK-NEXT: vpcmpeqw %xmm1, %xmm0, %k1
; CHECK-NEXT: kshiftrb $4, %k1, %k2
; CHECK-NEXT: vmovdqa64 32(%rdi), %ymm3 {%k2}
; CHECK-NEXT: vmovdqa64 (%rdi), %ymm2 {%k1}
; CHECK-NEXT: vmovdqa %ymm2, (%rdx)
; CHECK-NEXT: vmovdqa %ymm3, 32(%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq

```

```

%x = load <8 x i64>, <8 x i64>* %p
%y = load <8 x i64>, <8 x i64>* %q
%a = icmp eq <8 x i16> %s, %t
%b = select <8 x i1> %a, <8 x i64> %x, <8 x i64> %y
store <8 x i64> %b, <8 x i64>* %r
ret void
}

```

```

define dso_local void @vselect_split_v8i32_setcc(<8 x i32> %s, <8 x i32> %t, <8 x i64>* %p, <8 x i64>* %q, <8
x i64>* %r) "min-legal-vector-width"="256" {
; CHECK-LABEL: vselect_split_v8i32_setcc:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rsi), %ymm2
; CHECK-NEXT: vmovdqa 32(%rsi), %ymm3
; CHECK-NEXT: vpcmpeqd %ymm1,
%ymm0, %k1
; CHECK-NEXT: kshiftrb $4, %k1, %k2
; CHECK-NEXT: vmovdqa64 32(%rdi), %ymm3 {%k2}
; CHECK-NEXT: vmovdqa64 (%rdi), %ymm2 {%k1}
; CHECK-NEXT: vmovdqa %ymm2, (%rdx)
; CHECK-NEXT: vmovdqa %ymm3, 32(%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%x = load <8 x i64>, <8 x i64>* %p
%y = load <8 x i64>, <8 x i64>* %q
%a = icmp eq <8 x i32> %s, %t
%b = select <8 x i1> %a, <8 x i64> %x, <8 x i64> %y
store <8 x i64> %b, <8 x i64>* %r
ret void
}

```

```

define dso_local void @vselect_split_v16i8_setcc(<16 x i8> %s, <16 x i8> %t, <16 x i32>* %p, <16 x i32>* %q,
<16 x i32>* %r) "min-legal-vector-width"="256" {
; CHECK-LABEL: vselect_split_v16i8_setcc:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rsi), %ymm2
; CHECK-NEXT: vmovdqa 32(%rsi), %ymm3
; CHECK-NEXT: vpcmpeqb %xmm1, %xmm0, %k1
; CHECK-NEXT: kshiftrw $8, %k1, %k2
; CHECK-NEXT: vmovdqa32 32(%rdi), %ymm3 {%k2}
; CHECK-NEXT: vmovdqa32 (%rdi), %ymm2 {%k1}
; CHECK-NEXT: vmovdqa %ymm2,
(%rdx)
; CHECK-NEXT: vmovdqa %ymm3, 32(%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%x = load <16 x i32>, <16 x i32>* %p
%y = load <16 x i32>, <16 x i32>* %q

```

```

%a = icmp eq <16 x i8> %s, %t
%b = select <16 x i1> %a, <16 x i32> %x, <16 x i32> %y
store <16 x i32> %b, <16 x i32>* %r
ret void
}

define dso_local void @vselect_split_v16i16_setcc(<16 x i16> %s, <16 x i16> %t, <16 x i32>* %p, <16 x i32>*
%q, <16 x i32>* %r) "min-legal-vector-width"="256" {
; CHECK-LABEL: vselect_split_v16i16_setcc:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rsi), %ymm2
; CHECK-NEXT: vmovdqa 32(%rsi), %ymm3
; CHECK-NEXT: vpcmpeqw %ymm1, %ymm0, %k1
; CHECK-NEXT: kshiftrw $8, %k1, %k2
; CHECK-NEXT: vmovdqa32 32(%rdi), %ymm3 {%k2}
; CHECK-NEXT: vmovdqa32 (%rdi), %ymm2 {%k1}
; CHECK-NEXT: vmovdqa %ymm2, (%rdx)
; CHECK-NEXT: vmovdqa %ymm3, 32(%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%x = load <16 x i32>, <16 x i32>* %p
%y = load <16 x i32>,
<16 x i32>* %q
%a = icmp eq <16 x i16> %s, %t
%b = select <16 x i1> %a, <16 x i32> %x, <16 x i32> %y
store <16 x i32> %b, <16 x i32>* %r
ret void
}

define <16 x i8> @trunc_packus_v16i32_v16i8(<16 x i32>* %p) "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_packus_v16i32_v16i8:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rdi), %ymm0
; CHECK-NEXT: vpackusdw 32(%rdi), %ymm0, %ymm0
; CHECK-NEXT: vpermq {{.*#+}} ymm0 = ymm0[0,2,1,3]
; CHECK-NEXT: vpmovuswb %ymm0, %xmm0
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%a = load <16 x i32>, <16 x i32>* %p
%b = icmp slt <16 x i32> %a, <i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255>
%c = select <16 x i1> %b, <16 x i32> %a, <16 x i32> <i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255>
%d = icmp sgt <16 x i32> %c, zeroinitializer
%e = select <16 x i1> %d, <16 x i32> %c, <16 x i32> zeroinitializer
%f = trunc <16 x i32> %e to <16 x i8>
ret <16 x i8> %f
}

```



```

define dso_local void @trunc_packus_v16i32_v16i8_store(<16 x i32>* %p, <16 x i8>* %q) "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_packus_v16i32_v16i8_store:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rdi), %ymm0
; CHECK-NEXT: vpackusdw 32(%rdi), %ymm0, %ymm0
; CHECK-NEXT: vpermq {{.*#+}} ymm0 = ymm0[0,2,1,3]
; CHECK-NEXT: vpmovuswb %ymm0, (%rsi)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
    %a = load <16 x i32>, <16 x i32>* %p
    %b = icmp slt <16 x i32> %a, <i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255>
    %c = select <16 x i1> %b, <16 x i32> %a, <16 x i32> <i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255>
    %d = icmp sgt <16
    x i32> %c, zeroinitializer
    %e = select <16 x i1> %d, <16 x i32> %c, <16 x i32> zeroinitializer
    %f = trunc <16 x i32> %e to <16 x i8>
    store <16 x i8> %f, <16 x i8>* %q
    ret void
}

```

```

define <64 x i1> @v64i1_argument_return(<64 x i1> %x) "min-legal-vector-width"="256" {
; CHECK-LABEL: v64i1_argument_return:
; CHECK:      # %bb.0:
; CHECK-NEXT: retq
    ret <64 x i1> %x
}

```

```

define dso_local void @v64i1_shuffle(<64 x i8>* %x, <64 x i8>* %y) "min-legal-vector-width"="256" {
; CHECK-LABEL: v64i1_shuffle:
; CHECK:      # %bb.0: # %entry
; CHECK-NEXT: vmovdqa (%rdi), %ymm1
; CHECK-NEXT: vmovdqa 32(%rdi), %ymm0
; CHECK-NEXT: vptestnmb %ymm1, %ymm1, %k0
; CHECK-NEXT: kshiftrd $1, %k0, %k1
; CHECK-NEXT: kshiftrlq $63, %k0, %k2
; CHECK-NEXT: kshiftrq $62, %k2, %k2
; CHECK-NEXT: kshiftrlq $63, %k1, %k1
; CHECK-NEXT: kshiftrq $63, %k1, %k1
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-5, %rax
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1,
    %k1
; CHECK-NEXT: kshiftrd $3, %k0, %k2

```

```

; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $61, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-9, %rax
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $2, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $60, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-17, %rax
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $5, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $59, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-33, %rax
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $4, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $58, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-65, %rax
; CHECK-NEXT: kmovq %rax, %k2
;
CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $7, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $57, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-129, %rax
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $6, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $56, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-257, %rax # imm = 0xFEFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $9, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $55, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-513, %rax # imm = 0xFDFD
; CHECK-NEXT: kmovq %rax, %k2

```

```

; CHECK-NEXT:  kandq %k2, %k1, %k1
; CHECK-NEXT:  kshiftrd $8, %k0, %k2
; CHECK-NEXT:  kshiftrlq $63, %k2, %k2
; CHECK-NEXT:  kshiftrq $54, %k2, %k2
; CHECK-NEXT:  korq %k2, %k1, %k1
;
CHECK-NEXT:  movq $-1025, %rax # imm = 0xFBFF
; CHECK-NEXT:  kmovq %rax, %k2
; CHECK-NEXT:  kandq %k2, %k1, %k1
; CHECK-NEXT:  kshiftrd $11, %k0, %k2
; CHECK-NEXT:  kshiftrlq $63, %k2, %k2
; CHECK-NEXT:  kshiftrq $53, %k2, %k2
; CHECK-NEXT:  korq %k2, %k1, %k1
; CHECK-NEXT:  movq $-2049, %rax # imm = 0xF7FF
; CHECK-NEXT:  kmovq %rax, %k2
; CHECK-NEXT:  kandq %k2, %k1, %k1
; CHECK-NEXT:  kshiftrd $10, %k0, %k2
; CHECK-NEXT:  kshiftrlq $63, %k2, %k2
; CHECK-NEXT:  kshiftrq $52, %k2, %k2
; CHECK-NEXT:  korq %k2, %k1, %k1
; CHECK-NEXT:  movq $-4097, %rax # imm = 0xEFFF
; CHECK-NEXT:  kmovq %rax, %k2
; CHECK-NEXT:  kandq %k2, %k1, %k1
; CHECK-NEXT:  kshiftrd $13, %k0, %k2
; CHECK-NEXT:  kshiftrlq $63, %k2, %k2
; CHECK-NEXT:  kshiftrq $51, %k2, %k2
; CHECK-NEXT:  korq %k2, %k1, %k1
; CHECK-NEXT:  movq $-8193, %rax # imm = 0xDFFF
; CHECK-NEXT:  kmovq %rax, %k2
; CHECK-NEXT:  kandq %k2, %k1, %k1
; CHECK-NEXT:  kshiftrd $12, %k0, %k2
; CHECK-NEXT:  kshiftrlq $63, %k2, %k2
; CHECK-NEXT:  kshiftrq $50, %k2, %k2
; CHECK-NEXT:  korq %k2, %k1, %k1
; CHECK-NEXT:  movq $-16385, %rax # imm = 0xBFFF
; CHECK-NEXT:  kmovq %rax, %k2
; CHECK-NEXT:  kandq %k2, %k1, %k1
; CHECK-NEXT:  kshiftrd $15, %k0, %k2
; CHECK-NEXT:  kshiftrlq $63, %k2, %k2
; CHECK-NEXT:  kshiftrq $49, %k2, %k2
; CHECK-NEXT:  korq %k2, %k1, %k1
; CHECK-NEXT:  movq $-32769, %rax # imm = 0xFFFF7FFF
; CHECK-NEXT:  kmovq %rax, %k2
; CHECK-NEXT:  kandq %k2, %k1, %k1
; CHECK-NEXT:  kshiftrd $14, %k0, %k2
; CHECK-NEXT:  kshiftrlq $63, %k2, %k2
; CHECK-NEXT:  kshiftrq $48, %k2, %k2

```

```

; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-65537, %rax # imm = 0xFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $17, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $47, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-131073, %rax # imm = 0xFFFFDFFFF
;
CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $16, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $46, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-262145, %rax # imm = 0xFFFFBFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $19, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $45, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-524289, %rax # imm = 0xFFFF7FFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $18, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $44, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-1048577, %rax # imm = 0xFFEFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $21, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT:
kshiftrq $43, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-2097153, %rax # imm = 0xFFDFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $20, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $42, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-4194305, %rax # imm = 0xFFBFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1

```

```

; CHECK-NEXT: kshiftrd $23, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $41, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-8388609, %rax # imm = 0xFF7FFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $22, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $40, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-16777217, %rax # imm = 0xFEFFFFFF
; CHECK-NEXT: kmovq
%rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $25, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $39, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-33554433, %rax # imm = 0xFDFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $24, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $38, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-67108865, %rax # imm = 0xFBFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $27, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $37, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-134217729, %rax # imm = 0xF7FFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $26, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq
$36, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-268435457, %rax # imm = 0xEFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $29, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $35, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1

```

```

; CHECK-NEXT:  movq $-536870913, %rax # imm = 0xDFFFFFFF
; CHECK-NEXT:  kmovq %rax, %k2
; CHECK-NEXT:  kandq %k2, %k1, %k1
; CHECK-NEXT:  kshiftrd $28, %k0, %k2
; CHECK-NEXT:  kshiftrlq $63, %k2, %k2
; CHECK-NEXT:  kshiftrq $34, %k2, %k2
; CHECK-NEXT:  korq %k2, %k1, %k1
; CHECK-NEXT:  movq $-1073741825, %rax # imm = 0xBFFFFFFF
; CHECK-NEXT:  kmovq %rax, %k2
; CHECK-NEXT:  kandq %k2, %k1, %k1
; CHECK-NEXT:  kshiftrd $31, %k0, %k2
; CHECK-NEXT:  kshiftrlq $63, %k2, %k2
; CHECK-NEXT:  kshiftrq $33, %k2, %k2
; CHECK-NEXT:  korq %k2, %k1, %k1
; CHECK-NEXT:  movabsq $-2147483649, %rax # imm = 0xFFFFFFFF7FFFFFFF
; CHECK-NEXT:
    kmovq %rax, %k2
; CHECK-NEXT:  kandq %k2, %k1, %k2
; CHECK-NEXT:  vptestnmb %ymm0, %ymm0, %k1
; CHECK-NEXT:  kshiftrd $30, %k0, %k0
; CHECK-NEXT:  kshiftrlq $63, %k0, %k0
; CHECK-NEXT:  kshiftrq $32, %k0, %k0
; CHECK-NEXT:  korq %k0, %k2, %k0
; CHECK-NEXT:  movabsq $-4294967297, %rax # imm = 0xFFFFFFFFFFFFFFFF
; CHECK-NEXT:  kmovq %rax, %k2
; CHECK-NEXT:  kandq %k2, %k0, %k0
; CHECK-NEXT:  kshiftrd $1, %k1, %k2
; CHECK-NEXT:  kshiftrlq $63, %k2, %k2
; CHECK-NEXT:  kshiftrq $31, %k2, %k2
; CHECK-NEXT:  korq %k2, %k0, %k0
; CHECK-NEXT:  movabsq $-8589934593, %rax # imm = 0xFFFFFFFFDFFFFFFF
; CHECK-NEXT:  kmovq %rax, %k2
; CHECK-NEXT:  kandq %k2, %k0, %k0
; CHECK-NEXT:  kshiftrlq $63, %k1, %k2
; CHECK-NEXT:  kshiftrq $30, %k2, %k2
; CHECK-NEXT:  korq %k2, %k0, %k0
; CHECK-NEXT:  movabsq $-17179869185, %rax # imm = 0xFFFFFFFFBFFFFFFF
; CHECK-NEXT:  kmovq %rax, %k2
; CHECK-NEXT:  kandq %k2, %k0, %k0
; CHECK-NEXT:  kshiftrd $3, %k1, %k2
; CHECK-NEXT:
    kshiftrlq $63, %k2, %k2
; CHECK-NEXT:  kshiftrq $29, %k2, %k2
; CHECK-NEXT:  korq %k2, %k0, %k0
; CHECK-NEXT:  movabsq $-34359738369, %rax # imm = 0xFFFFFFFF7FFFFFFF
; CHECK-NEXT:  kmovq %rax, %k2
; CHECK-NEXT:  kandq %k2, %k0, %k0
; CHECK-NEXT:  kshiftrd $2, %k1, %k2

```

```

; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $28, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-68719476737, %rax # imm = 0xFFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $5, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $27, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-137438953473, %rax # imm = 0xFFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $4, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $26, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
;
CHECK-NEXT: movabsq $-274877906945, %rax # imm = 0xFFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $7, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $25, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-549755813889, %rax # imm = 0xFFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $6, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $24, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-1099511627777, %rax # imm = 0xFFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $9, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $23, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-2199023255553, %rax # imm = 0xFFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
;
CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $8, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $22, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-4398046511105, %rax # imm = 0xFFFFFFFFFFFFFFFF

```

```

; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $11, %k1, %k2
; CHECK-NEXT:    kshiftrlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $21, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-8796093022209, %rax # imm = 0xFFFFF7FFFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $10, %k1, %k2
; CHECK-NEXT:    kshiftrlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $20, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-17592186044417, %rax # imm = 0xFFFFEFFFFFFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $13, %k1, %k2
; CHECK-NEXT:    kshiftrlq
$63, %k2, %k2
; CHECK-NEXT:    kshiftrq $19, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-35184372088833, %rax # imm = 0xFFFFDFFFFFFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $12, %k1, %k2
; CHECK-NEXT:    kshiftrlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $18, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-70368744177665, %rax # imm = 0xFFFFBFFFFFFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $15, %k1, %k2
; CHECK-NEXT:    kshiftrlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $17, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-140737488355329, %rax # imm = 0xFFFF7FFFFFFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $14, %k1, %k2
; CHECK-NEXT:    kshiftrlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $16, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:
    movabsq $-281474976710657, %rax # imm = 0xFFFEFFFFFFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $17, %k1, %k2
; CHECK-NEXT:    kshiftrlq $63, %k2, %k2

```



```

; CHECK-NEXT: kshiftrq $15, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-562949953421313, %rax # imm = 0xFFFFDFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $16, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $14, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-1125899906842625, %rax # imm = 0xFFFBFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $19, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $13, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-2251799813685249, %rax # imm = 0xFFF7FFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq
%rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $18, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $12, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-4503599627370497, %rax # imm = 0xFFEFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $21, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $11, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-9007199254740993, %rax # imm = 0xFFDFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $20, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $10, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-18014398509481985, %rax # imm = 0xFFBFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $23, %k1, %k2
;
CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $9, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-36028797018963969, %rax # imm = 0xFF7FFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2

```

```

; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $22, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $8, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-72057594037927937, %rax # imm = 0xFEFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $25, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $7, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-144115188075855873, %rax # imm = 0xFDFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $24, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $6, %k2, %k2
; CHECK-NEXT:
    korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-288230376151711745, %rax # imm = 0xFBFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $27, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $5, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-576460752303423489, %rax # imm = 0xF7FFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $26, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $4, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-1152921504606846977, %rax # imm = 0xEEFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $29, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $3, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-2305843009213693953, %rax # imm =
0xDFFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $28, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $2, %k2, %k2

```

```

; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-4611686018427387905, %rax # imm = 0xBFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $31, %k1, %k2
; CHECK-NEXT: kshiftrlq $62, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $30, %k1, %k1
; CHECK-NEXT: kshiftrlq $1, %k0, %k0
; CHECK-NEXT: kshiftrq $1, %k0, %k0
; CHECK-NEXT: kshiftrlq $63, %k1, %k1
; CHECK-NEXT: korq %k1, %k0, %k1
; CHECK-NEXT: vmovdqu8 %ymm1, (%rsi) {%k1}
; CHECK-NEXT: kshiftrq $32, %k1, %k1
; CHECK-NEXT: vmovdqu8 %ymm0, 32(%rsi) {%k1}
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq

entry:
%a = load <64 x i8>, <64 x i8>* %x
%b = icmp eq <64 x i8> %a, zeroinitializer
%shuf = shufflevector
<64 x i1> %b, <64 x i1> undef, <64 x i32> <i32 1, i32 0, i32 3, i32 2, i32 5, i32 4, i32 7, i32 6, i32 9, i32 8, i32 11,
i32 10, i32 13, i32 12, i32 15, i32 14, i32 17, i32 16, i32 19, i32 18, i32 21, i32 20, i32 23, i32 22, i32 25, i32 24, i32
27, i32 26, i32 29, i32 28, i32 31, i32 30, i32 33, i32 32, i32 35, i32 34, i32 37, i32 36, i32 39, i32 38, i32 41, i32 40,
i32 43, i32 42, i32 45, i32 44, i32 47, i32 46, i32 49, i32 48, i32 51, i32 50, i32 53, i32 52, i32 55, i32 54, i32 57, i32
56, i32 59, i32 58, i32 61, i32 60, i32 63, i32 62>
call void @llvm.masked.store.v64i8.p0v64i8(<64 x i8> %a, <64 x i8>* %y, i32 1, <64 x i1> %shuf)
ret void
}

declare void @llvm.masked.store.v64i8.p0v64i8(<64 x i8>, <64 x i8>*, i32, <64 x i1>)

@mem64_dst = dso_local global i64 0, align 8
@mem64_src = dso_local global i64 0, align 8
define dso_local i32 @v64i1_inline_asm() "min-legal-vector-width"="256" {
; CHECK-LABEL: v64i1_inline_asm:
; CHECK:      # %bb.0:
; CHECK-NEXT: kmovq mem64_src(%rip),
%k0
; CHECK-NEXT: #APP
; CHECK-NEXT: #NO_APP
; CHECK-NEXT: kmovq %k0, mem64_dst(%rip)
; CHECK-NEXT: movl -{[0-9]+}}(%rsp), %eax
; CHECK-NEXT: retq
%1 = alloca i32, align 4
%2 = load i64, i64* @mem64_src, align 8
%3 = call i64 @asm "", "=k,k,~{dirflag},~{fpsr},~{flags}"(i64 %2)
store i64 %3, i64* @mem64_dst, align 8
%4 = load i32, i32* %1, align 4

```

```
ret i32 %4
}
```

```
define dso_local void @cmp_v8i64_sext(<8 x i64>* %xptr, <8 x i64>* %yptr, <8 x i64>* %zptr) "min-legal-
vector-width"="256" {
; CHECK-LABEL: cmp_v8i64_sext:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rsi), %ymm0
; CHECK-NEXT: vmovdqa 32(%rsi), %ymm1
; CHECK-NEXT: vpcmpgtq 32(%rdi), %ymm1, %ymm1
; CHECK-NEXT: vpcmpgtq (%rdi), %ymm0, %ymm0
; CHECK-NEXT: vmovdqa %ymm0, (%rdx)
; CHECK-NEXT: vmovdqa %ymm1, 32(%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%x = load <8 x i64>, <8 x i64>* %xptr
%y = load <8 x i64>, <8 x i64>* %yptr
%cmp = icmp slt <8 x i64> %x, %y
%ext
= sext <8 x i1> %cmp to <8 x i64>
store <8 x i64> %ext, <8 x i64>* %zptr
ret void
}
```

```
define dso_local void @cmp_v8i64_zext(<8 x i64>* %xptr, <8 x i64>* %yptr, <8 x i64>* %zptr) "min-legal-
vector-width"="256" {
; CHECK-LABEL: cmp_v8i64_zext:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rsi), %ymm0
; CHECK-NEXT: vmovdqa 32(%rsi), %ymm1
; CHECK-NEXT: vpcmpgtq 32(%rdi), %ymm1, %ymm1
; CHECK-NEXT: vpcmpgtq (%rdi), %ymm0, %ymm0
; CHECK-NEXT: vpsrlq $63, %ymm1, %ymm1
; CHECK-NEXT: vpsrlq $63, %ymm0, %ymm0
; CHECK-NEXT: vmovdqa %ymm0, (%rdx)
; CHECK-NEXT: vmovdqa %ymm1, 32(%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%x = load <8 x i64>, <8 x i64>* %xptr
%y = load <8 x i64>, <8 x i64>* %yptr
%cmp = icmp slt <8 x i64> %x, %y
%ext = zext <8 x i1> %cmp to <8 x i64>
store <8 x i64> %ext, <8 x i64>* %zptr
ret void
}
```

```
define <16 x i8> @var_rotate_v16i8(<16 x i8> %a, <16 x i8> %b) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: var_rotate_v16i8:
```

```
;
CHECK:    # %bb.0:
; CHECK-NEXT:  vpadd {{\.?LCPI[0-9]+_[0-9]+}}(%rip){1to4}, %xmm1, %xmm1
; CHECK-NEXT:  vpxor %xmm2, %xmm2, %xmm2
; CHECK-NEXT:  vpunpckhbw {{.*#+}} xmm2 =
xmm1[8],xmm2[8],xmm1[9],xmm2[9],xmm1[10],xmm2[10],xmm1[11],xmm2[11],xmm1[12],xmm2[12],xmm1[13],
xmm2[13],xmm1[14],xmm2[14],xmm1[15],xmm2[15]
; CHECK-NEXT:  vpunpckhbw {{.*#+}} xmm3 = xmm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15]
; CHECK-NEXT:  vpsllvw %xmm2, %xmm3, %xmm2
; CHECK-NEXT:  vpsrlw $8, %xmm2, %xmm2
; CHECK-NEXT:  vpunpcklbw {{.*#+}} xmm0 = xmm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7]
; CHECK-NEXT:  vpmovzxbw {{.*#+}} xmm1 =
xmm1[0],zero,xmm1[1],zero,xmm1[2],zero,xmm1[3],zero,xmm1[4],zero,xmm1[5],zero,xmm1[6],zero,xmm1[7],zer
o
; CHECK-NEXT:  vpsllvw %xmm1, %xmm0, %xmm0
; CHECK-NEXT:  vpsrlw $8, %xmm0, %xmm0
; CHECK-NEXT:  vpackuswb %xmm2, %xmm0, %xmm0
; CHECK-NEXT:  retq
%b8 = sub <16 x i8> <i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8>, %b
i8 8, i8 8, i8 8>, %b
%shl = shl <16 x i8> %a, %b
%lshr = lshr <16 x i8> %a, %b8
%or = or <16 x i8> %shl, %lshr
ret <16 x i8> %or
}
```

```
define <32 x i8> @var_rotate_v32i8(<32 x i8> %a, <32 x i8> %b) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: var_rotate_v32i8:
; CHECK:    # %bb.0:
; CHECK-NEXT:  vpadd {{\.?LCPI[0-9]+_[0-9]+}}(%rip){1to8}, %ymm1, %ymm1
; CHECK-NEXT:  vpxor %xmm2, %xmm2, %xmm2
; CHECK-NEXT:  vpunpckhbw {{.*#+}} ymm3 =
ymm1[8],ymm2[8],ymm1[9],ymm2[9],ymm1[10],ymm2[10],ymm1[11],ymm2[11],ymm1[12],ymm2[12],ymm1[13],
ymm2[13],ymm1[14],ymm2[14],ymm1[15],ymm2[15],ymm1[24],ymm2[24],ymm1[25],ymm2[25],ymm1[26],ymm
2[26],ymm1[27],ymm2[27],ymm1[28],ymm2[28],ymm1[29],ymm2[29],ymm1[30],ymm2[30],ymm1[31],ymm2[
31]
; CHECK-NEXT:  vpunpckhbw {{.*#+}} ymm4 =
ymm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-NEXT:  vpsllvw %ymm3, %ymm4, %ymm3
; CHECK-NEXT:  vpsrlw $8, %ymm3, %ymm3
; CHECK-NEXT:  vpunpcklbw
{{.*#+}} ymm1 =
ymm1[0],ymm2[0],ymm1[1],ymm2[1],ymm1[2],ymm2[2],ymm1[3],ymm2[3],ymm1[4],ymm2[4],ymm1[5],ymm2[
5],ymm1[6],ymm2[6],ymm1[7],ymm2[7],ymm1[16],ymm2[16],ymm1[17],ymm2[17],ymm1[18],ymm2[18],ymm1[
19],ymm2[19],ymm1[20],ymm2[20],ymm1[21],ymm2[21],ymm1[22],ymm2[22],ymm1[23],ymm2[23]
; CHECK-NEXT:  vpunpcklbw {{.*#+}} ymm0 =
ymm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-NEXT:  vpsllvw %ymm1, %ymm0, %ymm0
```

```
; CHECK-NEXT: vpsrlw $8, %ymm0, %ymm0
; CHECK-NEXT: vpackuswb %ymm3, %ymm0, %ymm0
; CHECK-NEXT: retq
%b8 = sub <32 x i8> <i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8>, %b
%shl = shl <32 x i8> %a, %b
%lshr = lshr <32 x i8> %a, %b8
%or = or <32 x i8> %shl, %lshr
ret <32 x i8> %or
}
```

```
define <32 x i8> @splatvar_rotate_v32i8(<32 x i8> %a, <32 x i8> %b) nounwind "min-legal-vector-width"="256"
{
; CHECK-LABEL: splatvar_rotate_v32i8:
; CHECK:      # %bb.0:
; CHECK-NEXT: vpunpckhbw {{.*#+}} ymm2 =
ymm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-NEXT: vpand {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %xmm1, %xmm1
; CHECK-NEXT: vpsllw %xmm1, %ymm2, %ymm2
; CHECK-NEXT: vpsrlw $8, %ymm2, %ymm2
; CHECK-NEXT: vpunpcklbw {{.*#+}} ymm0 =
ymm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-NEXT: vpsllw %xmm1, %ymm0, %ymm0
; CHECK-NEXT: vpsrlw $8, %ymm0, %ymm0
; CHECK-NEXT: vpackuswb %ymm2, %ymm0, %ymm0
; CHECK-NEXT: retq
%splat = shufflevector <32 x i8> %b, <32 x i8> undef, <32 x i32> zeroinitializer
%splat8 = sub <32 x i8> <i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8>, %splat
%shl = shl <32 x i8>
%a, %splat
%lshr = lshr <32 x i8> %a, %splat8
%or = or <32 x i8> %shl, %lshr
ret <32 x i8> %or
}
```

```
define <32 x i8> @constant_rotate_v32i8(<32 x i8> %a) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: constant_rotate_v32i8:
; CHECK:      # %bb.0:
; CHECK-NEXT: vpunpckhbw {{.*#+}} ymm1 =
ymm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-NEXT: vpsllvw {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %ymm1, %ymm1
; CHECK-NEXT: vpsrlw $8, %ymm1, %ymm1
; CHECK-NEXT: vpunpcklbw {{.*#+}} ymm0 =
ymm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-NEXT: vpsllvw {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %ymm0, %ymm0
; CHECK-NEXT: vpsrlw $8, %ymm0, %ymm0
; CHECK-NEXT: vpackuswb %ymm1, %ymm0, %ymm0
```

```
; CHECK-NEXT:    retq
%shl = shl <32 x i8> %a, <i8 0, i8 1, i8 2, i8 3, i8 4, i8 5, i8 6, i8 7, i8 8, i8 7, i8 6, i8 5, i8 4, i8 3, i8 2, i8 1, i8 0, i8 1, i8 2, i8 3, i8 4, i8 5, i8 6, i8 7, i8 8, i8 7, i8 6, i8 5, i8 4, i8 3, i8 2, i8 1>
%lshr = lshr <32 x i8> %a, <i8 8, i8 7, i8 6, i8 5, i8 4, i8 3, i8 2, i8 1, i8 0, i8 1, i8 2, i8 3, i8 4, i8 5, i8 6, i8 7, i8 8, i8 7, i8 6, i8 5, i8 4, i8 3, i8 2, i8 1, i8 0, i8 1, i8 2, i8 3, i8 4, i8 5, i8 6, i8 7>
%or = or <32 x i8> %shl, %lshr
ret <32 x i8> %or
}
```

```
define <32 x i8> @splatconstant_rotate_v32i8(<32 x i8> %a) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: splatconstant_rotate_v32i8:
; CHECK:      # %bb.0:
; CHECK-NEXT:  vpsllw $4, %ymm0, %ymm1
; CHECK-NEXT:  vpsrlw $4, %ymm0, %ymm0
; CHECK-NEXT:  vpternlogd $216, {{\.?LCPI[0-9]+\_[0-9]+\}}(%rip){1to8}, %ymm1, %ymm0
; CHECK-NEXT:  retq
%shl = shl <32 x i8> %a, <i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4>
%lshr = lshr <32 x i8> %a, <i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4>
%or = or <32 x i8> %shl, %lshr
ret <32 x i8> %or
}
```

```
define <32 x i8> @splatconstant_rotate_mask_v32i8(<32 x i8> %a) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: splatconstant_rotate_mask_v32i8:
; CHECK:      # %bb.0:
; CHECK-NEXT:  vpsllw $4, %ymm0, %ymm1
; CHECK-NEXT:  vpsrlw $4, %ymm0, %ymm0
; CHECK-NEXT:  vpternlogd $216, {{\.?LCPI[0-9]+\_[0-9]+\}}(%rip){1to8}, %ymm1, %ymm0
; CHECK-NEXT:  vpandd {{\.?LCPI[0-9]+\_[0-9]+\}}(%rip){1to8}, %ymm0, %ymm0
; CHECK-NEXT:  retq
%shl = shl <32 x i8> %a, <i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4>
%lshr = lshr <32 x i8> %a, <i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4>
%rmask = and <32 x i8> %lshr, <i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55>
%lmask = and <32 x i8> %shl, <i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33>
%or = or <32 x i8> %lmask, %rmask
ret <32 x i8> %or
}
```

```
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefixes=CHECK,RV64I
; RUN: llc -mtriple=riscv64 -mattr=+zbs -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefixes=CHECK,RV64ZBS
```

```
define signext i32 @bclr_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: bclr_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sllw a1, a2, a1
; RV64I-NEXT:  not a1, a1
; RV64I-NEXT:  and a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bclr_i32:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  andi a1, a1, 31
; RV64ZBS-NEXT:  bclr a0, a0, a1
; RV64ZBS-NEXT:  sext.w a0, a0
; RV64ZBS-NEXT:  ret
%and = and i32 %b, 31
%shl = shl nuw i32 1, %and
%neg = xor i32 %shl, -1
%and1 = and i32 %neg, %a
ret i32 %and1
}
```

```
define signext i32 @bclr_i32_no_mask(i32 signext %a,
i32 signext %b) nounwind {
; RV64I-LABEL: bclr_i32_no_mask:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sllw a1, a2, a1
; RV64I-NEXT:  not a1, a1
; RV64I-NEXT:  and a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bclr_i32_no_mask:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  bclr a0, a0, a1
; RV64ZBS-NEXT:  sext.w a0, a0
; RV64ZBS-NEXT:  ret
%shl = shl i32 1, %b
%neg = xor i32 %shl, -1
%and1 = and i32 %neg, %a
ret i32 %and1
}
```



```

define signext i32 @bclr_i32_load(ptr %p, i32 signext %b) nounwind {
; RV64I-LABEL: bclr_i32_load:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lw a0, 0(a0)
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sllw a1, a2, a1
; RV64I-NEXT:  not a1, a1
; RV64I-NEXT:  and a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bclr_i32_load:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  lw a0, 0(a0)
; RV64ZBS-NEXT:  bclr a0, a0, a1
; RV64ZBS-NEXT:  sext.w a0, a0
; RV64ZBS-NEXT:  ret
%a = load i32, ptr %p
%shl = shl i32 1, %b
%neg
= xor i32 %shl, -1
%and1 = and i32 %neg, %a
ret i32 %and1
}

```

```

define i64 @bclr_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: bclr_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sll a1, a2, a1
; RV64I-NEXT:  not a1, a1
; RV64I-NEXT:  and a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bclr_i64:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  bclr a0, a0, a1
; RV64ZBS-NEXT:  ret
%and = and i64 %b, 63
%shl = shl nuw i64 1, %and
%neg = xor i64 %shl, -1
%and1 = and i64 %neg, %a
ret i64 %and1
}

```

```

define i64 @bclr_i64_no_mask(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: bclr_i64_no_mask:
; RV64I:      # %bb.0:

```

```

; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sll a1, a2, a1
; RV64I-NEXT:  not a1, a1
; RV64I-NEXT:  and a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bclr_i64_no_mask:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  bclr a0, a0, a1
; RV64ZBS-NEXT:  ret
%shl = shl i64 1, %b
%neg = xor i64 %shl, -1
%and1 = and i64 %neg, %a
ret i64
%and1
}

```

```

define signext i32 @bset_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: bset_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sllw a1, a2, a1
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bset_i32:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  andi a1, a1, 31
; RV64ZBS-NEXT:  bset a0, a0, a1
; RV64ZBS-NEXT:  sext.w a0, a0
; RV64ZBS-NEXT:  ret
%and = and i32 %b, 31
%shl = shl nuw i32 1, %and
%or = or i32 %shl, %a
ret i32 %or
}

```

```

define signext i32 @bset_i32_no_mask(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: bset_i32_no_mask:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sllw a1, a2, a1
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bset_i32_no_mask:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  bset a0, a0, a1

```

```

; RV64ZBS-NEXT:  sext.w a0, a0
; RV64ZBS-NEXT:  ret
%shl = shl i32 1, %b
%or = or i32 %shl, %a
ret i32 %or
}

define signext i32
@bset_i32_load(ptr %p, i32 signext %b) nounwind {
; RV64I-LABEL: bset_i32_load:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lw a0, 0(a0)
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sllw a1, a2, a1
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bset_i32_load:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  lw a0, 0(a0)
; RV64ZBS-NEXT:  bset a0, a0, a1
; RV64ZBS-NEXT:  sext.w a0, a0
; RV64ZBS-NEXT:  ret
%a = load i32, ptr %p
%shl = shl i32 1, %b
%or = or i32 %shl, %a
ret i32 %or
}

```

; We can use bsetw for  $1 \ll x$  by setting the first source to zero.

```

define signext i32 @bset_i32_zero(i32 signext %a) nounwind {
; RV64I-LABEL: bset_i32_zero:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 1
; RV64I-NEXT:  sllw a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bset_i32_zero:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bset a0, zero, a0
; RV64ZBS-NEXT:  sext.w a0, a0
; RV64ZBS-NEXT:  ret
%shl = shl i32 1, %a
ret i32 %shl
}

```

```

define i64 @bset_i64(i64 %a,

```

```

i64 %b) nounwind {
; RV64I-LABEL: bset_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT: li a2, 1
; RV64I-NEXT: sll a1, a2, a1
; RV64I-NEXT: or a0, a1, a0
; RV64I-NEXT: ret
;
; RV64ZBS-LABEL: bset_i64:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT: bset a0, a0, a1
; RV64ZBS-NEXT: ret
%conv = and i64 %b, 63
%shl = shl nuw i64 1, %conv
%or = or i64 %shl, %a
ret i64 %or
}

```

```

define i64 @bset_i64_no_mask(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: bset_i64_no_mask:
; RV64I:      # %bb.0:
; RV64I-NEXT: li a2, 1
; RV64I-NEXT: sll a1, a2, a1
; RV64I-NEXT: or a0, a1, a0
; RV64I-NEXT: ret
;
; RV64ZBS-LABEL: bset_i64_no_mask:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT: bset a0, a0, a1
; RV64ZBS-NEXT: ret
%shl = shl i64 1, %b
%or = or i64 %shl, %a
ret i64 %or
}

```

; We can use bsetw for  $1 \ll x$  by setting the first source to zero.

```

define signext i64 @bset_i64_zero(i64 signext %a) nounwind {
; RV64I-LABEL: bset_i64_zero:
; RV64I:      # %bb.0:
; RV64I-NEXT: li
a1, 1
; RV64I-NEXT: sll a0, a1, a0
; RV64I-NEXT: ret
;
; RV64ZBS-LABEL: bset_i64_zero:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT: bset a0, zero, a0
; RV64ZBS-NEXT: ret

```

```

%shl = shl i64 1, %a
ret i64 %shl
}

define signext i32 @binv_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: binv_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sllw a1, a2, a1
; RV64I-NEXT:  xor a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: binv_i32:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  andi a1, a1, 31
; RV64ZBS-NEXT:  binv a0, a0, a1
; RV64ZBS-NEXT:  sext.w a0, a0
; RV64ZBS-NEXT:  ret
%and = and i32 %b, 31
%shl = shl nuw i32 1, %and
%xor = xor i32 %shl, %a
ret i32 %xor
}

define signext i32 @binv_i32_no_mask(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: binv_i32_no_mask:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sllw a1, a2, a1
; RV64I-NEXT:  xor a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: binv_i32_no_mask:
;
RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  binv a0, a0, a1
; RV64ZBS-NEXT:  sext.w a0, a0
; RV64ZBS-NEXT:  ret
%shl = shl i32 1, %b
%xor = xor i32 %shl, %a
ret i32 %xor
}

define signext i32 @binv_i32_load(ptr %p, i32 signext %b) nounwind {
; RV64I-LABEL: binv_i32_load:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lw a0, 0(a0)
; RV64I-NEXT:  li a2, 1

```

```

; RV64I-NEXT:  sllw a1, a2, a1
; RV64I-NEXT:  xor a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: binv_i32_load:
; RV64ZBS:     # %bb.0:
; RV64ZBS-NEXT:  lw a0, 0(a0)
; RV64ZBS-NEXT:  binv a0, a0, a1
; RV64ZBS-NEXT:  sext.w a0, a0
; RV64ZBS-NEXT:  ret
%a = load i32, ptr %p
%shl = shl i32 1, %b
%xor = xor i32 %shl, %a
ret i32 %xor
}

define i64 @binv_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: binv_i64:
; RV64I:     # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sll a1, a2, a1
; RV64I-NEXT:  xor a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: binv_i64:
; RV64ZBS:     # %bb.0:
;
RV64ZBS-NEXT:  binv a0, a0, a1
; RV64ZBS-NEXT:  ret
%conv = and i64 %b, 63
%shl = shl nuw i64 1, %conv
%xor = xor i64 %shl, %a
ret i64 %xor
}

define i64 @binv_i64_no_mask(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: binv_i64_no_mask:
; RV64I:     # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sll a1, a2, a1
; RV64I-NEXT:  xor a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: binv_i64_no_mask:
; RV64ZBS:     # %bb.0:
; RV64ZBS-NEXT:  binv a0, a0, a1
; RV64ZBS-NEXT:  ret
%shl = shl nuw i64 1, %b

```

```

%xor = xor i64 %shl, %a
ret i64 %xor
}

```

```

define signext i32 @bext_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: bext_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srlw a0, a0, a1
; RV64I-NEXT:  andi a0, a0, 1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bext_i32:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  andi a1, a1, 31
; RV64ZBS-NEXT:  bext a0, a0, a1
; RV64ZBS-NEXT:  ret
%and = and i32 %b, 31
%shr = lshr i32 %a, %and
%and1 = and i32
%shr, 1
ret i32 %and1
}

```

```

define signext i32 @bext_i32_no_mask(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: bext_i32_no_mask:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srlw a0, a0, a1
; RV64I-NEXT:  andi a0, a0, 1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bext_i32_no_mask:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bext a0, a0, a1
; RV64ZBS-NEXT:  ret
%shr = lshr i32 %a, %b
%and1 = and i32 %shr, 1
ret i32 %and1
}

```

; This gets previous converted to (i1 (truncate (srl X, Y))). Make sure we are  
; able to use bext.

```

define void @bext_i32_trunc(i32 signext %0, i32 signext %1) {
; RV64I-LABEL: bext_i32_trunc:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srlw a0, a0, a1
; RV64I-NEXT:  andi a0, a0, 1
; RV64I-NEXT:  beqz a0, .LBB19_2
; RV64I-NEXT:  # %bb.1:

```

```

; RV64I-NEXT:  ret
; RV64I-NEXT:  .LBB19_2:
; RV64I-NEXT:  tail bar
;
; RV64ZBS-LABEL: bext_i32_trunc:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  bext a0, a0, a1
; RV64ZBS-NEXT:  beqz a0, .LBB19_2
; RV64ZBS-NEXT:
; RV64ZBS-NEXT:  # %bb.1:
; RV64ZBS-NEXT:  ret
; RV64ZBS-NEXT:  .LBB19_2:
; RV64ZBS-NEXT:  tail bar
%3 = shl i32 1, %1
%4 = and i32 %3, %0
%5 = icmp eq i32 %4, 0
br i1 %5, label %6, label %7

6:                                     ; preds = %2
tail call void @bar()
br label %7

7:                                     ; preds = %6, %2
ret void
}

declare void @bar()

define i64 @bext_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: bext_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srl a0, a0, a1
; RV64I-NEXT:  andi a0, a0, 1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bext_i64:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  bext a0, a0, a1
; RV64ZBS-NEXT:  ret
%conv = and i64 %b, 63
%shr = lshr i64 %a, %conv
%and1 = and i64 %shr, 1
ret i64 %and1
}

define i64 @bext_i64_no_mask(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: bext_i64_no_mask:
; RV64I:      # %bb.0:

```



```

; RV64I-NEXT:  srl a0, a0, a1
; RV64I-NEXT:  andi a0, a0, 1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL:
bext_i64_no_mask:
; RV64ZBS:     # %bb.0:
; RV64ZBS-NEXT: bext a0, a0, a1
; RV64ZBS-NEXT: ret
%shr = lshr i64 %a, %b
%and1 = and i64 %shr, 1
ret i64 %and1
}

```

```

define signext i32 @bexti_i32(i32 signext %a) nounwind {
; RV64I-LABEL: bexti_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT: slli a0, a0, 58
; RV64I-NEXT: srli a0, a0, 63
; RV64I-NEXT: ret
;
; RV64ZBS-LABEL: bexti_i32:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT: bexti a0, a0, 5
; RV64ZBS-NEXT: ret
%shr = lshr i32 %a, 5
%and = and i32 %shr, 1
ret i32 %and
}

```

```

define i64 @bexti_i64(i64 %a) nounwind {
; RV64I-LABEL: bexti_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT: slli a0, a0, 58
; RV64I-NEXT: srli a0, a0, 63
; RV64I-NEXT: ret
;
; RV64ZBS-LABEL: bexti_i64:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT: bexti a0, a0, 5
; RV64ZBS-NEXT: ret
%shr = lshr i64 %a, 5
%and = and i64 %shr, 1
ret i64 %and
}

```

```

define signext i32 @bexti_i32_cmp(i32 signext %a) nounwind {
; CHECK-LABEL: bexti_i32_cmp:

```

```

;
CHECK:    # %bb.0:
; CHECK-NEXT:  andi a0, a0, 32
; CHECK-NEXT:  snez a0, a0
; CHECK-NEXT:  ret
%and = and i32 %a, 32
%cmp = icmp ne i32 %and, 0
%zext = zext i1 %cmp to i32
ret i32 %zext
}

define i64 @bexti_i64_cmp(i64 %a) nounwind {
; RV64I-LABEL: bexti_i64_cmp:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 58
; RV64I-NEXT:  srli a0, a0, 63
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bexti_i64_cmp:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bexti a0, a0, 5
; RV64ZBS-NEXT:  ret
%and = and i64 %a, 32
%cmp = icmp ne i64 %and, 0
%zext = zext i1 %cmp to i64
ret i64 %zext
}

define signext i32 @bclri_i32_10(i32 signext %a) nounwind {
; CHECK-LABEL: bclri_i32_10:
; CHECK:    # %bb.0:
; CHECK-NEXT:  andi a0, a0, -1025
; CHECK-NEXT:  ret
%and = and i32 %a, -1025
ret i32 %and
}

define signext i32 @bclri_i32_11(i32 signext %a) nounwind {
; RV64I-LABEL: bclri_i32_11:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a1, 1048575
; RV64I-NEXT:  addiw a1,
a1, 2047
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bclri_i32_11:
; RV64ZBS:    # %bb.0:

```

```

; RV64ZBS-NEXT:  bclri a0, a0, 11
; RV64ZBS-NEXT:  ret
%and = and i32 %a, -2049
ret i32 %and
}

define signext i32 @bclri_i32_30(i32 signext %a) nounwind {
; RV64I-LABEL: bclri_i32_30:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a1, 786432
; RV64I-NEXT:  addiw a1, a1, -1
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bclri_i32_30:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bclri a0, a0, 30
; RV64ZBS-NEXT:  ret
%and = and i32 %a, -1073741825
ret i32 %and
}

define signext i32 @bclri_i32_31(i32 signext %a) nounwind {
; RV64I-LABEL: bclri_i32_31:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 33
; RV64I-NEXT:  srli a0, a0, 33
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bclri_i32_31:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bclri a0, a0, 31
; RV64ZBS-NEXT:  sext.w a0, a0
; RV64ZBS-NEXT:  ret
%and = and i32 %a,
-2147483649
ret i32 %and
}

define i64 @bclri_i64_10(i64 %a) nounwind {
; CHECK-LABEL: bclri_i64_10:
; CHECK:      # %bb.0:
; CHECK-NEXT:  andi a0, a0, -1025
; CHECK-NEXT:  ret
%and = and i64 %a, -1025
ret i64 %and
}

```

```

define i64 @bclri_i64_11(i64 %a) nounwind {
; RV64I-LABEL: bclri_i64_11:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a1, 1048575
; RV64I-NEXT:  addiw a1, a1, 2047
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bclri_i64_11:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  bclri a0, a0, 11
; RV64ZBS-NEXT:  ret
%and = and i64 %a, -2049
ret i64 %and
}

```

```

define i64 @bclri_i64_30(i64 %a) nounwind {
; RV64I-LABEL: bclri_i64_30:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a1, 786432
; RV64I-NEXT:  addiw a1, a1, -1
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bclri_i64_30:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  bclri a0, a0, 30
; RV64ZBS-NEXT:  ret
%and = and i64 %a, -1073741825
ret i64 %and
}

```

```

define
i64 @bclri_i64_31(i64 %a) nounwind {
; RV64I-LABEL: bclri_i64_31:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a1, 524288
; RV64I-NEXT:  addi a1, a1, -1
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bclri_i64_31:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  bclri a0, a0, 31
; RV64ZBS-NEXT:  ret
%and = and i64 %a, -2147483649
ret i64 %and
}

```

```

define i64 @bclri_i64_62(i64 %a) nounwind {
; RV64I-LABEL: bclri_i64_62:
; RV64I:      # %bb.0:
; RV64I-NEXT: li a1, -1
; RV64I-NEXT: slli a1, a1, 62
; RV64I-NEXT: addi a1, a1, -1
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: ret
;
; RV64ZBS-LABEL: bclri_i64_62:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT: bclri a0, a0, 62
; RV64ZBS-NEXT: ret
%and = and i64 %a, -4611686018427387905
ret i64 %and
}

```

```

define i64 @bclri_i64_63(i64 %a) nounwind {
; RV64I-LABEL: bclri_i64_63:
; RV64I:      # %bb.0:
; RV64I-NEXT: slli a0, a0, 1
; RV64I-NEXT: srli a0, a0, 1
; RV64I-NEXT: ret
;
; RV64ZBS-LABEL:
bclri_i64_63:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT: bclri a0, a0, 63
; RV64ZBS-NEXT: ret
%and = and i64 %a, -9223372036854775809
ret i64 %and
}

```

```

define i64 @bclri_i64_large0(i64 %a) nounwind {
; RV64I-LABEL: bclri_i64_large0:
; RV64I:      # %bb.0:
; RV64I-NEXT: lui a1, 1044480
; RV64I-NEXT: addiw a1, a1, -256
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: ret
;
; RV64ZBS-LABEL: bclri_i64_large0:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT: andi a0, a0, -256
; RV64ZBS-NEXT: bclri a0, a0, 24
; RV64ZBS-NEXT: ret
%and = and i64 %a, -16777472

```

```

ret i64 %and
}

define i64 @bclri_i64_large1(i64 %a) nounwind {
; RV64I-LABEL: bclri_i64_large1:
; RV64I:      # %bb.0:
; RV64I-NEXT: lui a1, 1044464
; RV64I-NEXT: addiw a1, a1, -1
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: ret
;
; RV64ZBS-LABEL: bclri_i64_large1:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT: bclri a0, a0, 16
; RV64ZBS-NEXT: bclri a0, a0, 24
; RV64ZBS-NEXT: ret
%and = and i64 %a,
-16842753
ret i64 %and
}

define signext i32 @bseti_i32_10(i32 signext %a) nounwind {
; CHECK-LABEL: bseti_i32_10:
; CHECK:      # %bb.0:
; CHECK-NEXT: ori a0, a0, 1024
; CHECK-NEXT: ret
%or = or i32 %a, 1024
ret i32 %or
}

define signext i32 @bseti_i32_11(i32 signext %a) nounwind {
; RV64I-LABEL: bseti_i32_11:
; RV64I:      # %bb.0:
; RV64I-NEXT: li a1, 1
; RV64I-NEXT: slliw a1, a1, 11
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: ret
;
; RV64ZBS-LABEL: bseti_i32_11:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT: bseti a0, a0, 11
; RV64ZBS-NEXT: ret
%or = or i32 %a, 2048
ret i32 %or
}

define signext i32 @bseti_i32_30(i32 signext %a) nounwind {
; RV64I-LABEL: bseti_i32_30:

```

```

; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a1, 262144
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bseti_i32_30:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bseti a0, a0, 30
; RV64ZBS-NEXT:  ret
%or = or i32 %a, 1073741824
ret i32 %or
}

```

```

define signext i32
@bseti_i32_31(i32 signext %a) nounwind {
; RV64I-LABEL: bseti_i32_31:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a1, 524288
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bseti_i32_31:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bseti a0, a0, 31
; RV64ZBS-NEXT:  sext.w a0, a0
; RV64ZBS-NEXT:  ret
%or = or i32 %a, 2147483648
ret i32 %or
}

```

```

define i64 @bseti_i64_10(i64 %a) nounwind {
; CHECK-LABEL: bseti_i64_10:
; CHECK:    # %bb.0:
; CHECK-NEXT:  ori a0, a0, 1024
; CHECK-NEXT:  ret
%or = or i64 %a, 1024
ret i64 %or
}

```

```

define i64 @bseti_i64_11(i64 %a) nounwind {
; RV64I-LABEL: bseti_i64_11:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a1, 1
; RV64I-NEXT:  slli a1, a1, 11
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bseti_i64_11:

```

```

; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bseti a0, a0, 11
; RV64ZBS-NEXT:  ret
%or = or i64 %a, 2048
ret i64 %or
}

define i64 @bseti_i64_30(i64 %a) nounwind {
; RV64I-LABEL: bseti_i64_30:
;
RV64I:    # %bb.0:
; RV64I-NEXT:  lui a1, 262144
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bseti_i64_30:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bseti a0, a0, 30
; RV64ZBS-NEXT:  ret
%or = or i64 %a, 1073741824
ret i64 %or
}

define i64 @bseti_i64_31(i64 %a) nounwind {
; RV64I-LABEL: bseti_i64_31:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a1, 1
; RV64I-NEXT:  slli a1, a1, 31
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bseti_i64_31:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bseti a0, a0, 31
; RV64ZBS-NEXT:  ret
%or = or i64 %a, 2147483648
ret i64 %or
}

define i64 @bseti_i64_62(i64 %a) nounwind {
; RV64I-LABEL: bseti_i64_62:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a1, 1
; RV64I-NEXT:  slli a1, a1, 62
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bseti_i64_62:

```



```

; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bseti a0, a0, 62
; RV64ZBS-NEXT:  ret
%or = or i64 %a,
4611686018427387904
ret i64 %or
}

define i64 @bseti_i64_63(i64 %a) nounwind {
; RV64I-LABEL: bseti_i64_63:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a1, -1
; RV64I-NEXT:  slli a1, a1, 63
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bseti_i64_63:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bseti a0, a0, 63
; RV64ZBS-NEXT:  ret
%or = or i64 %a, 9223372036854775808
ret i64 %or
}

define signext i32 @binvi_i32_10(i32 signext %a) nounwind {
; CHECK-LABEL: binvi_i32_10:
; CHECK:    # %bb.0:
; CHECK-NEXT:  xori a0, a0, 1024
; CHECK-NEXT:  ret
%xor = xor i32 %a, 1024
ret i32 %xor
}

define signext i32 @binvi_i32_11(i32 signext %a) nounwind {
; RV64I-LABEL: binvi_i32_11:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a1, 1
; RV64I-NEXT:  slliw a1, a1, 11
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: binvi_i32_11:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  binvi a0, a0, 11
; RV64ZBS-NEXT:  ret
%xor = xor i32 %a,
2048
ret i32 %xor
}

```

```

}

define signext i32 @binvi_i32_30(i32 signext %a) nounwind {
; RV64I-LABEL: binvi_i32_30:
; RV64I:      # %bb.0:
; RV64I-NEXT: lui a1, 262144
; RV64I-NEXT: xor a0, a0, a1
; RV64I-NEXT: ret
;
; RV64ZBS-LABEL: binvi_i32_30:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT: binvi a0, a0, 30
; RV64ZBS-NEXT: ret
%xor = xor i32 %a, 1073741824
ret i32 %xor
}

```

```

define signext i32 @binvi_i32_31(i32 signext %a) nounwind {
; RV64I-LABEL: binvi_i32_31:
; RV64I:      # %bb.0:
; RV64I-NEXT: lui a1, 524288
; RV64I-NEXT: xor a0, a0, a1
; RV64I-NEXT: ret
;
; RV64ZBS-LABEL: binvi_i32_31:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT: binvi a0, a0, 31
; RV64ZBS-NEXT: sext.w a0, a0
; RV64ZBS-NEXT: ret
%xor = xor i32 %a, 2147483648
ret i32 %xor
}

```

```

define i64 @binvi_i64_10(i64 %a) nounwind {
; CHECK-LABEL: binvi_i64_10:
; CHECK:      # %bb.0:
; CHECK-NEXT: xori a0, a0, 1024
; CHECK-NEXT: ret
%xor = xor i64 %a, 1024
ret i64 %xor
}

```

```

define i64 @binvi_i64_11(i64
%a) nounwind {
; RV64I-LABEL: binvi_i64_11:
; RV64I:      # %bb.0:
; RV64I-NEXT: li a1, 1
; RV64I-NEXT: slli a1, a1, 11

```

```

; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: binvi_i64_11:
; RV64ZBS:     # %bb.0:
; RV64ZBS-NEXT:  binvi a0, a0, 11
; RV64ZBS-NEXT:  ret
%xor = xor i64 %a, 2048
ret i64 %xor
}

define i64 @binvi_i64_30(i64 %a) nounwind {
; RV64I-LABEL: binvi_i64_30:
; RV64I:       # %bb.0:
; RV64I-NEXT:  lui a1, 262144
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: binvi_i64_30:
; RV64ZBS:     # %bb.0:
; RV64ZBS-NEXT:  binvi a0, a0, 30
; RV64ZBS-NEXT:  ret
%xor = xor i64 %a, 1073741824
ret i64 %xor
}

define i64 @binvi_i64_31(i64 %a) nounwind {
; RV64I-LABEL: binvi_i64_31:
; RV64I:       # %bb.0:
; RV64I-NEXT:  li a1, 1
; RV64I-NEXT:  slli a1, a1, 31
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: binvi_i64_31:
; RV64ZBS:     # %bb.0:
; RV64ZBS-NEXT:  binvi
a0, a0, 31
; RV64ZBS-NEXT:  ret
%xor = xor i64 %a, 2147483648
ret i64 %xor
}

define i64 @binvi_i64_62(i64 %a) nounwind {
; RV64I-LABEL: binvi_i64_62:
; RV64I:       # %bb.0:
; RV64I-NEXT:  li a1, 1
; RV64I-NEXT:  slli a1, a1, 62

```

```

; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: binvi_i64_62:
; RV64ZBS:     # %bb.0:
; RV64ZBS-NEXT:  binvi a0, a0, 62
; RV64ZBS-NEXT:  ret
%xor = xor i64 %a, 4611686018427387904
ret i64 %xor
}

define i64 @binvi_i64_63(i64 %a) nounwind {
; RV64I-LABEL: binvi_i64_63:
; RV64I:       # %bb.0:
; RV64I-NEXT:  li a1, -1
; RV64I-NEXT:  slli a1, a1, 63
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: binvi_i64_63:
; RV64ZBS:     # %bb.0:
; RV64ZBS-NEXT:  binvi a0, a0, 63
; RV64ZBS-NEXT:  ret
%xor = xor i64 %a, 9223372036854775808
ret i64 %xor
}

define i64 @xor_i64_large(i64 %a) nounwind {
; RV64I-LABEL: xor_i64_large:
; RV64I:       # %bb.0:
; RV64I-NEXT:  li a1, 1
; RV64I-NEXT:  slli
a1, a1, 32
; RV64I-NEXT:  addi a1, a1, 1
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: xor_i64_large:
; RV64ZBS:     # %bb.0:
; RV64ZBS-NEXT:  binvi a0, a0, 0
; RV64ZBS-NEXT:  binvi a0, a0, 32
; RV64ZBS-NEXT:  ret
%xor = xor i64 %a, 4294967297
ret i64 %xor
}

define i64 @xor_i64_4099(i64 %a) nounwind {
; RV64I-LABEL: xor_i64_4099:

```

```

; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a1, 1
; RV64I-NEXT:  addiw a1, a1, 3
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: xor_i64_4099:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  xori a0, a0, 3
; RV64ZBS-NEXT:  binvi a0, a0, 12
; RV64ZBS-NEXT:  ret
%xor = xor i64 %a, 4099
ret i64 %xor
}

```

```

define i64 @xor_i64_96(i64 %a) nounwind {
; CHECK-LABEL: xor_i64_96:
; CHECK:      # %bb.0:
; CHECK-NEXT:  xori a0, a0, 96
; CHECK-NEXT:  ret
%xor = xor i64 %a, 96
ret i64 %xor
}

```

```

define i64 @or_i64_large(i64 %a) nounwind {
; RV64I-LABEL: or_i64_large:
; RV64I:      # %bb.0:
; RV64I-NEXT:
    li a1, 1
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  addi a1, a1, 1
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: or_i64_large:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bseti a0, a0, 0
; RV64ZBS-NEXT:  bseti a0, a0, 32
; RV64ZBS-NEXT:  ret
%or = or i64 %a, 4294967297
ret i64 %or
}

```

```

define i64 @xor_i64_66901(i64 %a) nounwind {
; RV64I-LABEL: xor_i64_66901:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a1, 16
; RV64I-NEXT:  addiw a1, a1, 1365

```

```

; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: xor_i64_66901:
; RV64ZBS:     # %bb.0:
; RV64ZBS-NEXT: xori a0, a0, 1365
; RV64ZBS-NEXT: binvi a0, a0, 16
; RV64ZBS-NEXT: ret
%xor = xor i64 %a, 66901
ret i64 %xor
}

define i64 @or_i64_4099(i64 %a) nounwind {
; RV64I-LABEL: or_i64_4099:
; RV64I:      # %bb.0:
; RV64I-NEXT: lui a1, 1
; RV64I-NEXT: addiw a1, a1, 3
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: ret
;
; RV64ZBS-LABEL: or_i64_4099:
; RV64ZBS:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT: ori a0, a0, 3
; RV64ZBS-NEXT: bseti a0, a0, 12
; RV64ZBS-NEXT: ret
%or = or i64 %a, 4099
ret i64 %or
}

define i64 @or_i64_96(i64 %a) nounwind {
; CHECK-LABEL: or_i64_96:
; CHECK:      # %bb.0:
; CHECK-NEXT: ori a0, a0, 96
; CHECK-NEXT: ret
%or = or i64 %a, 96
ret i64 %or
}

define i64 @or_i64_66901(i64 %a) nounwind {
; RV64I-LABEL: or_i64_66901:
; RV64I:      # %bb.0:
; RV64I-NEXT: lui a1, 16
; RV64I-NEXT: addiw a1, a1, 1365
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: ret
;
; RV64ZBS-LABEL: or_i64_66901:

```

```

; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  ori a0, a0, 1365
; RV64ZBS-NEXT:  bseti a0, a0, 16
; RV64ZBS-NEXT:  ret
%or = or i64 %a, 66901
ret i64 %or
}

define signext i32 @bset_trailing_ones_i32_mask(i32 signext %a) nounwind {
; RV64I-LABEL: bset_trailing_ones_i32_mask:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a1, -1
; RV64I-NEXT:  sllw a0, a1, a0
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  ret
;
;
RV64ZBS-LABEL: bset_trailing_ones_i32_mask:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  andi a0, a0, 31
; RV64ZBS-NEXT:  bset a0, zero, a0
; RV64ZBS-NEXT:  addiw a0, a0, -1
; RV64ZBS-NEXT:  ret
%and = and i32 %a, 31
%shift = shl nsw i32 -1, %and
%not = xor i32 %shift, -1
ret i32 %not
}

define signext i32 @bset_trailing_ones_i32_no_mask(i32 signext %a) nounwind {
; RV64I-LABEL: bset_trailing_ones_i32_no_mask:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a1, -1
; RV64I-NEXT:  sllw a0, a1, a0
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  ret
;
RV64ZBS-LABEL: bset_trailing_ones_i32_no_mask:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bset a0, zero, a0
; RV64ZBS-NEXT:  addiw a0, a0, -1
; RV64ZBS-NEXT:  ret
%shift = shl nsw i32 -1, %a
%not = xor i32 %shift, -1
ret i32 %not
}

define signext i64 @bset_trailing_ones_i64_mask(i64 signext %a) nounwind {

```

```

; RV64I-LABEL: bset_trailing_ones_i64_mask:
; RV64I:      # %bb.0:
; RV64I-NEXT: li a1, -1
; RV64I-NEXT: sll
a0, a1, a0
; RV64I-NEXT: not a0, a0
; RV64I-NEXT: ret
;
; RV64ZBS-LABEL: bset_trailing_ones_i64_mask:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT: bset a0, zero, a0
; RV64ZBS-NEXT: addi a0, a0, -1
; RV64ZBS-NEXT: ret
%and = and i64 %a, 63
%shift = shl nsw i64 -1, %and
%not = xor i64 %shift, -1
ret i64 %not
}

define signext i64 @bset_trailing_ones_i64_no_mask(i64 signext %a) nounwind {
; RV64I-LABEL: bset_trailing_ones_i64_no_mask:
; RV64I:      # %bb.0:
; RV64I-NEXT: li a1, -1
; RV64I-NEXT: sll a0, a1, a0
; RV64I-NEXT: not a0, a0
; RV64I-NEXT: ret
;
; RV64ZBS-LABEL: bset_trailing_ones_i64_no_mask:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT: bset a0, zero, a0
; RV64ZBS-NEXT: addi a0, a0, -1
; RV64ZBS-NEXT: ret
%shift = shl nsw i64 -1, %a
%not = xor i64 %shift, -1
ret i64 %not
}

```

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
 Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.



"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition,

"submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each

Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

#### ---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions  
of this Software are embedded into an Object form of such source code, you  
may redistribute such embedded portions in such Object form without complying  
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled  
forms of this Software with  
software that is licensed under the GPLv2 ("Combined Software") and if a  
court of competent jurisdiction determines that the patent provision (Section  
3), the indemnity provision (Section 9) or other Section of the License  
conflicts with the conditions of the GPLv2, you may retroactively and  
prospectively choose to deem waived or otherwise exclude such Section(s) of  
the License, but only in their entirety and only with respect to the Combined  
Software.

=====  
Software from third parties included in the LLVM Project:  
=====

The LLVM Project contains third party software which is under different license  
terms. All such code will be identified clearly using at least one of two  
mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or  
`LICENSE` file at the top containing the specific  
license and restrictions  
which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every  
file.

=====  
Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):  
=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2003-2019 University of Illinois at Urbana-Champaign.  
All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

<http://llvm.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do

so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- \* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition,

"submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and



- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

## APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

#### ---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions  
of this Software are embedded into an Object form of such source code, you  
may redistribute such embedded portions in such Object form without complying  
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled  
forms of this Software with  
software that is licensed under the GPLv2 ("Combined Software") and if a  
court of competent jurisdiction determines that the patent provision (Section  
3), the indemnity provision (Section 9) or other Section of the License  
conflicts with the conditions of the GPLv2, you may retroactively and  
prospectively choose to deem waived or otherwise exclude such Section(s) of  
the License, but only in their entirety and only with respect to the Combined  
Software.

=====  
Software from third parties included in the LLVM Project:  
=====

The LLVM Project contains third party software which is under different license  
terms. All such code will be identified clearly using at least one of two  
mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or  
`LICENSE` file at the top containing the specific  
license and restrictions  
which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every  
file.

=====  
Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):

=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2010 Apple Inc.  
All rights reserved.

Developed by:

LLDB Team

<http://lldb.llvm.org/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- \* Neither the names of the LLDB Team, copyright holders, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

-----

clang-tidy High-Integrity C++ Files

-----

All clang-tidy files are licensed under the same terms as the rest of the LLVM project with the following additions:

Any file referencing a High-Integrity C++ Coding guideline:

HIC++ Coding Standard as created by PRQA.

Please see

<https://www.perforce.com/resources/qac/high-integrity-cpp-coding-standard/conditions-of-use>  
for more information.

ISC LICENSE

This license is approved by the OSI and FSF as GPL-compatible.

<http://opensource.org/licenses/isc-license.txt>

Copyright (c) 2013-2014, Pexpect development team

Copyright (c) 2012, Noah Spurrier <[noah@noah.org](mailto:noah@noah.org)>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

```
; RUN: opt -passes=consthoist -S -o - %s | FileCheck %s
target datalayout = "e-m:e-p:32:32-i64:64-v128:64:128-a:0:32-n32-S64"
target triple = "thumbv6m-none--musleabi"
```

```
; Check that for i8 type, the maximum legal offset is 31.
; Also check that an constant used as value to be stored rather than
; pointer in a store instruction is hoisted.
; CHECK: foo_i8
; CHECK-DAG: %[[C1:const[0-9]?]] = bitcast i32 805874720 to i32
; CHECK-DAG: %[[C2:const[0-9]?]] = bitcast i32 805874688 to i32
; CHECK-DAG: %[[C3:const[0-9]?]] = bitcast i32 805873720 to i32
; CHECK-DAG: %[[C4:const[0-9]?]] = bitcast i32 805873688 to i32
; CHECK: %0 = inttoptr i32 %[[C2]] to ptr
; CHECK-NEXT: %1 = load volatile i8, ptr %0
; CHECK-NEXT: %[[M1:const_mat[0-9]?]] = add i32 %[[C2]], 4
; CHECK-NEXT: %2 = inttoptr i32 %[[M1]] to ptr
; CHECK-NEXT: %3 = load volatile i8, ptr %2
; CHECK-NEXT: %[[M2:const_mat[0-9]?]] = add i32 %[[C2]], 31
; CHECK-NEXT: %4 = inttoptr i32 %[[M2]] to ptr
; CHECK-NEXT: %5 = load volatile
i8, ptr %4
; CHECK-NEXT: %6 = inttoptr i32 %[[C1]] to ptr
; CHECK-NEXT: %7 = load volatile i8, ptr %6
; CHECK-NEXT: %[[M3:const_mat[0-9]?]] = add i32 %[[C1]], 7
```

```

; CHECK-NEXT: %8 = inttoptr i32 %[[M3]] to ptr
; CHECK-NEXT: %9 = load volatile i8, ptr %8
; CHECK-NEXT: %10 = inttoptr i32 %[[C4]] to ptr
; CHECK-NEXT: store i8 %9, ptr %10
; CHECK-NEXT: %[[M4:const_mat[0-9]?]] = add i32 %[[C4]], 31
; CHECK-NEXT: %11 = inttoptr i32 %[[M4]] to ptr
; CHECK-NEXT: store i8 %7, ptr %11
; CHECK-NEXT: %12 = inttoptr i32 %[[C3]] to ptr
; CHECK-NEXT: store i8 %5, ptr %12
; CHECK-NEXT: %[[M5:const_mat[0-9]?]] = add i32 %[[C3]], 7
; CHECK-NEXT: %13 = inttoptr i32 %[[M5]] to ptr
; CHECK-NEXT: store i8 %3, ptr %13
; CHECK-NEXT: %[[M6:const_mat[0-9]?]] = add i32 %[[C1]], 80
; CHECK-NEXT: %14 = inttoptr i32 %[[M6]] to ptr
; CHECK-NEXT: store ptr %14, ptr @goo

```

@goo = global ptr undef

```

define void @foo_i8() {
entry:
  %0 = load volatile i8, ptr inttoptr (i32 805874688 to ptr)
  %1 = load volatile
i8, ptr inttoptr (i32 805874692 to ptr)
  %2 = load volatile i8, ptr inttoptr (i32 805874719 to ptr)
  %3 = load volatile i8, ptr inttoptr (i32 805874720 to ptr)
  %4 = load volatile i8, ptr inttoptr (i32 805874727 to ptr)
  store i8 %4, ptr inttoptr(i32 805873688 to ptr)
  store i8 %3, ptr inttoptr(i32 805873719 to ptr)
  store i8 %2, ptr inttoptr(i32 805873720 to ptr)
  store i8 %1, ptr inttoptr(i32 805873727 to ptr)
  store ptr inttoptr(i32 805874800 to ptr), ptr @goo
  ret void
}

```

```

; Check that for i16 type, the maximum legal offset is 62.
; CHECK: foo_i16
; CHECK-DAG: %[[C1:const[0-9]?]] = bitcast i32 805874752 to i32
; CHECK-DAG: %[[C2:const[0-9]?]] = bitcast i32 805874688 to i32
; CHECK: %0 = inttoptr i32 %[[C2]] to ptr
; CHECK-NEXT: %1 = load volatile i16, ptr %0, align 2
; CHECK-NEXT: %[[M1:const_mat[0-9]?]] = add i32 %[[C2]], 4
; CHECK-NEXT: %2 = inttoptr i32 %[[M1]] to ptr
; CHECK-NEXT: %3 = load volatile i16, ptr %2, align 2
; CHECK-NEXT: %[[M2:const_mat[0-9]?]] = add
i32 %[[C2]], 32
; CHECK-NEXT: %4 = inttoptr i32 %[[M2]] to ptr
; CHECK-NEXT: %5 = load volatile i16, ptr %4, align 2
; CHECK-NEXT: %[[M3:const_mat[0-9]?]] = add i32 %[[C2]], 62

```

```
; CHECK-NEXT: %6 = inttoptr i32 %[[M3]] to ptr
; CHECK-NEXT: %7 = load volatile i16, ptr %6, align 2
; CHECK-NEXT: %8 = inttoptr i32 %[[C1]] to ptr
; CHECK-NEXT: %9 = load volatile i16, ptr %8, align 2
; CHECK-NEXT: %[[M4:const_mat[0-9]?]] = add i32 %[[C1]], 22
; CHECK-NEXT: %10 = inttoptr i32 %[[M4]] to ptr
; CHECK-NEXT: %11 = load volatile i16, ptr %10, align 2
```

```
define void @foo_i16() {
entry:
  %0 = load volatile i16, ptr inttoptr (i32 805874688 to ptr), align 2
  %1 = load volatile i16, ptr inttoptr (i32 805874692 to ptr), align 2
  %2 = load volatile i16, ptr inttoptr (i32 805874720 to ptr), align 2
  %3 = load volatile i16, ptr inttoptr (i32 805874750 to ptr), align 2
  %4 = load volatile i16, ptr inttoptr (i32 805874752 to ptr), align 2
  %5 = load volatile i16, ptr inttoptr (i32 805874774 to ptr), align
  2
  ret void
}
```

```
; Check that for i32 type, the maximum legal offset is 124.
; CHECK: foo_i32
; CHECK-DAG: %[[C1:const[0-9]?]] = bitcast i32 805874816 to i32
; CHECK-DAG: %[[C2:const[0-9]?]] = bitcast i32 805874688 to i32
; CHECK: %0 = inttoptr i32 %[[C2]] to ptr
; CHECK-NEXT: %1 = load volatile i32, ptr %0, align 4
; CHECK-NEXT: %[[M1:const_mat[0-9]?]] = add i32 %[[C2]], 4
; CHECK-NEXT: %2 = inttoptr i32 %[[M1]] to ptr
; CHECK-NEXT: %3 = load volatile i32, ptr %2, align 4
; CHECK-NEXT: %[[M2:const_mat[0-9]?]] = add i32 %[[C2]], 124
; CHECK-NEXT: %4 = inttoptr i32 %[[M2]] to ptr
; CHECK-NEXT: %5 = load volatile i32, ptr %4, align 4
; CHECK-NEXT: %6 = inttoptr i32 %[[C1]] to ptr
; CHECK-NEXT: %7 = load volatile i32, ptr %6, align 4
; CHECK-NEXT: %[[M3:const_mat[0-9]?]] = add i32 %[[C1]], 8
; CHECK-NEXT: %8 = inttoptr i32 %[[M3]] to ptr
; CHECK-NEXT: %9 = load volatile i32, ptr %8, align 4
; CHECK-NEXT: %[[M4:const_mat[0-9]?]] = add i32 %[[C1]], 12
; CHECK-NEXT: %10 = inttoptr
  i32 %[[M4]] to ptr
; CHECK-NEXT: %11 = load volatile i32, ptr %10, align 4
```

```
define void @foo_i32() {
entry:
  %0 = load volatile i32, ptr inttoptr (i32 805874688 to ptr), align 4
  %1 = load volatile i32, ptr inttoptr (i32 805874692 to ptr), align 4
  %2 = load volatile i32, ptr inttoptr (i32 805874812 to ptr), align 4
  %3 = load volatile i32, ptr inttoptr (i32 805874816 to ptr), align 4
```

```

%4 = load volatile i32, ptr inttoptr (i32 805874824 to ptr), align 4
%5 = load volatile i32, ptr inttoptr (i32 805874828 to ptr), align 4
ret void
}
; RUN: llc -march=hexagon < %s
; REQUIRES: asserts

; The two loads based on %struct.0, loading two different data types
; cause LSR to assume type "void" for the memory type. This would then
; cause an assert in isLegalAddressingMode. Make sure we no longer crash.

```

```
target triple = "hexagon"
```

```

%struct.0 = type { ptr, i8, %union.anon.0 }
%union.anon.0 = type { ptr }

```

```

define hidden fastcc void @fred() unnamed_addr #0 {
entry:
br i1 undef, label %while.end, label %while.body.lr.ph

```

```

while.body.lr.ph:                ; preds = %entry
br label %while.body

```

```

while.body:                      ; preds = %exit.2, %while.body.lr.ph
%lsr.iv = phi ptr [ %cgep22, %exit.2 ], [ undef, %while.body.lr.ph ]
switch i32 undef, label %exit [
  i32 1, label %sw.bb.i
  i32 2, label %sw.bb3.i
]

```

```

sw.bb.i:                        ; preds = %while.body
unreachable

```

```

sw.bb3.i:                      ; preds = %while.body
unreachable

```

```

exit:                            ; preds = %while.body
switch i32 undef, label %exit.2 [
  i32 1, label %sw.bb.i17
  i32 2, label %sw.bb3.i20
]

```

```

sw.bb.i17:                     ; preds = %exit
%0 = load i32, ptr %lsr.iv, align 4
unreachable

```

```

sw.bb3.i20:                   ; preds = %exit

```



```
%1 = load ptr, ptr %lsr.iv, align 4
```

```
unreachable
```

```
exit.2:                                ; preds = %exit  
%cgep22 = getelementptr %struct.0, ptr %lsr.iv, i32 1  
br label %while.body
```

```
while.end:                            ; preds = %entry  
ret void  
}
```

```
attributes #0 = { nounwind optsize "target-cpu"="hexagonv55" }
```

```
; NOTE: Assertions have been autogenerated by utils/update_test_checks.py UTC_ARGS: --include-generated-funcs
```

```
; RUN: opt -S -passes=verify,iroutliner -ir-outlining-no-cost < %s | FileCheck %s
```

```
; FIXME: Disabled run-line after running into some turbulence with debuginfo-iterators
```

```
; run: opt -S -passes=verify,iroutliner -ir-outlining-no-cost < %s --try-experimental-debuginfo-iterators | FileCheck %s
```

```
; This test checks that debug info is recognized as able to be extracted along  
; with the other instructions, but is not included in the consolidated function.
```

```
define void @function1() !dbg !6 {  
entry:  
  %a = alloca i32, align 4, !dbg !17  
  call void @llvm.dbg.value(metadata ptr %a, metadata !9, metadata !DIExpression()), !dbg !17  
  %b = alloca i32, align 4, !dbg !18  
  call void @llvm.dbg.value(metadata ptr %b, metadata !11, metadata !DIExpression()), !dbg !18  
  %c = alloca i32, align 4, !dbg !19  
  call void @llvm.dbg.value(metadata ptr %c, metadata !12, metadata !DIExpression()), !dbg !19  
  
  store i32 2, ptr %a, align 4, !dbg !20  
  store i32 3, ptr %b, align 4, !dbg !21  
  store i32 4, ptr %c, align 4, !dbg !22  
  %al = load i32, ptr %a, align 4, !dbg !23  
  call void @llvm.dbg.value(metadata i32 %al, metadata !13, metadata !DIExpression()), !dbg !23  
  %bl = load i32, ptr %b, align 4, !dbg !24  
  call void @llvm.dbg.value(metadata i32 %bl, metadata !15, metadata !DIExpression()), !dbg !24  
  %cl = load i32, ptr %c, align 4, !dbg !25  
  call void @llvm.dbg.value(metadata i32 %cl, metadata !16, metadata !DIExpression()), !dbg !25  
  ret void, !dbg !26  
}
```

```
define void @function2() !dbg !27 {  
entry:  
  %a = alloca i32, align 4, !dbg !35  
  call void @llvm.dbg.value(metadata ptr %a, metadata !29, metadata !DIExpression()), !dbg !35  
  %b = alloca i32, align 4, !dbg !36
```

```

call void @llvm.dbg.value(metadata ptr %b, metadata !30, metadata !DIExpression()), !dbg !36
%c = alloca i32, align 4, !dbg !37
call void @llvm.dbg.value(metadata ptr %c, metadata !31, metadata !DIExpression()), !dbg
!37
store i32 2, ptr %a, align 4, !dbg !38
store i32 3, ptr %b, align 4, !dbg !39
store i32 4, ptr %c, align 4, !dbg !40
%al = load i32, ptr %a, align 4, !dbg !41
call void @llvm.dbg.value(metadata i32 %al, metadata !32, metadata !DIExpression()), !dbg !41
%bl = load i32, ptr %b, align 4, !dbg !42
call void @llvm.dbg.value(metadata i32 %bl, metadata !33, metadata !DIExpression()), !dbg !42
%cl = load i32, ptr %c, align 4, !dbg !43
call void @llvm.dbg.value(metadata i32 %cl, metadata !34, metadata !DIExpression()), !dbg !43
ret void, !dbg !44
}

```

```

; Function Attrs: nounwind readnone speculatable willreturn
declare void @llvm.dbg.value(metadata, metadata, metadata) #0

```

```

attributes #0 = { nounwind readnone speculatable willreturn }

```

```

!llvm.dbg.cu = !{!0}
!llvm.debugify = !{!3, !4}
!llvm.module.flags = !{!5}

```

```

!0 = distinct !DICompileUnit(language: DW_LANG_C, file: !1, producer: "debugify", isOptimized: true,
runtimeVersion: 0, emissionKind: FullDebug, enums: !2)
!1 = !DIFile(filename:
"legal-debug.ll", directory: "")
!2 = !{}
!3 = !{i32 20}
!4 = !{i32 12}
!5 = !{i32 2, !"Debug Info Version", i32 3}
!6 = distinct !DISubprogram(name: "function1", linkageName: "function1", scope: null, file: !1, line: 1, type: !7,
scopeLine: 1, spFlags: DISPFlagDefinition | DISPFlagOptimized, unit: !0, retainedNodes: !8)
!7 = !DISubroutineType(types: !2)
!8 = !{!9, !11, !12, !13, !15, !16}
!9 = !DILocalVariable(name: "1", scope: !6, file: !1, line: 1, type: !10)
!10 = !DIBasicType(name: "ty64", size: 64, encoding: DW_ATE_unsigned)
!11 = !DILocalVariable(name: "2", scope: !6, file: !1, line: 2, type: !10)
!12 = !DILocalVariable(name: "3", scope: !6, file: !1, line: 3, type: !10)
!13 = !DILocalVariable(name: "4", scope: !6, file: !1, line: 7, type: !14)
!14 = !DIBasicType(name: "ty32", size: 32, encoding: DW_ATE_unsigned)
!15 = !DILocalVariable(name: "5", scope: !6, file: !1, line: 8, type: !14)
!16 = !DILocalVariable(name: "6", scope: !6, file: !1, line: 9, type: !14)
!17

```

```

= !DILocation(line: 1, column: 1, scope: !6)
!18 = !DILocation(line: 2, column: 1, scope: !6)
!19 = !DILocation(line: 3, column: 1, scope: !6)
!20 = !DILocation(line: 4, column: 1, scope: !6)
!21 = !DILocation(line: 5, column: 1, scope: !6)
!22 = !DILocation(line: 6, column: 1, scope: !6)
!23 = !DILocation(line: 7, column: 1, scope: !6)
!24 = !DILocation(line: 8, column: 1, scope: !6)
!25 = !DILocation(line: 9, column: 1, scope: !6)
!26 = !DILocation(line: 10, column: 1, scope: !6)
!27 = distinct !DISubprogram(name: "function2", linkageName: "function2", scope: null, file: !1, line: 11, type: !7,
scopeLine: 11, spFlags: DISPFlagDefinition | DISPFlagOptimized, unit: !0, retainedNodes: !28)
!28 = !{!29, !30, !31, !32, !33, !34}
!29 = !DILocalVariable(name: "7", scope: !27, file: !1, line: 11, type: !10)
!30 = !DILocalVariable(name: "8", scope: !27, file: !1, line: 12, type: !10)
!31 = !DILocalVariable(name: "9", scope: !27, file: !1, line: 13, type: !10)
!32 = !DILocalVariable(name: "10",
scope: !27, file: !1, line: 17, type: !14)
!33 = !DILocalVariable(name: "11", scope: !27, file: !1, line: 18, type: !14)
!34 = !DILocalVariable(name: "12", scope: !27, file: !1, line: 19, type: !14)
!35 = !DILocation(line: 11, column: 1, scope: !27)
!36 = !DILocation(line: 12, column: 1, scope: !27)
!37 = !DILocation(line: 13, column: 1, scope: !27)
!38 = !DILocation(line: 14, column: 1, scope: !27)
!39 = !DILocation(line: 15, column: 1, scope: !27)
!40 = !DILocation(line: 16, column: 1, scope: !27)
!41 = !DILocation(line: 17, column: 1, scope: !27)
!42 = !DILocation(line: 18, column: 1, scope: !27)
!43 = !DILocation(line: 19, column: 1, scope: !27)
!44 = !DILocation(line: 20, column: 1, scope: !27)
; CHECK-LABEL: @function1(
; CHECK-NEXT: entry:
; CHECK-NEXT: [[A:%.*]] = alloca i32, align 4, !dbg [[DBG17:![0-9]+]]
; CHECK-NEXT: call void @llvm.dbg.value(metadata ptr [[A]], metadata [[META9:![0-9]+]], metadata
!DIExpression()), !dbg [[DBG17]]
; CHECK-NEXT: [[B:%.*]] = alloca
i32, align 4, !dbg [[DBG18:![0-9]+]]
; CHECK-NEXT: call void @llvm.dbg.value(metadata ptr [[B]], metadata [[META11:![0-9]+]], metadata
!DIExpression()), !dbg [[DBG18]]
; CHECK-NEXT: [[C:%.*]] = alloca i32, align 4, !dbg [[DBG19:![0-9]+]]
; CHECK-NEXT: call void @llvm.dbg.value(metadata ptr [[C]], metadata [[META12:![0-9]+]], metadata
!DIExpression()), !dbg [[DBG19]]
; CHECK-NEXT: call void @outlined_ir_func_0(ptr [[A]], ptr [[B]], ptr [[C]]), !dbg [[DBG20:![0-9]+]]
; CHECK-NEXT: ret void, !dbg [[DBG21:![0-9]+]]
;
;
; CHECK-LABEL: @function2(
; CHECK-NEXT: entry:

```

```

; CHECK-NEXT: [[A:%.*]] = alloca i32, align 4, !dbg [[DBG30:[0-9]+]]
; CHECK-NEXT: call void @llvm.dbg.value(metadata ptr [[A]], metadata [[META24:[0-9]+]], metadata
!DIExpression()), !dbg [[DBG30]]
; CHECK-NEXT: [[B:%.*]] = alloca i32, align 4, !dbg [[DBG31:[0-9]+]]
; CHECK-NEXT: call void @llvm.dbg.value(metadata ptr [[B]], metadata [[META25:[0-9]+]], metadata
!DIExpression()), !dbg [[DBG31]]
;
CHECK-NEXT: [[C:%.*]] = alloca i32, align 4, !dbg [[DBG32:[0-9]+]]
; CHECK-NEXT: call void @llvm.dbg.value(metadata ptr [[C]], metadata [[META26:[0-9]+]], metadata
!DIExpression()), !dbg [[DBG32]]
; CHECK-NEXT: call void @outlined_ir_func_0(ptr [[A]], ptr [[B]], ptr [[C]]), !dbg [[DBG33:[0-9]+]]
; CHECK-NEXT: ret void, !dbg [[DBG34:[0-9]+]]
;
;
; CHECK: @outlined_ir_func_0(ptr [[TMP0:%.*]], ptr [[TMP1:%.*]], ptr [[TMP2:%.*]])
; CHECK: entry_to_outline:
; CHECK-NEXT: store i32 2, ptr [[TMP0]], align 4
; CHECK-NEXT: store i32 3, ptr [[TMP1]], align 4
; CHECK-NEXT: store i32 4, ptr [[TMP2]], align 4
; CHECK-NEXT: [[AL:%.*]] = load i32, ptr [[TMP0]], align 4
; CHECK-NEXT: [[BL:%.*]] = load i32, ptr [[TMP1]], align 4
; CHECK-NEXT: [[CL:%.*]] = load i32, ptr [[TMP2]], align 4
; CHECK-NEXT: br label [[ENTRY_AFTER_OUTLINE_EXITSTUB:%.*]]
;
; RUN: opt < %s -passes=argpromotion -S | FileCheck %s

; CHECK-LABEL: define i32 @foo() #0 {
; CHECK-NEXT:    %.val = load <32 x half>, ptr undef, align 4
; CHECK-NEXT:    call void @bar(<32 x half> %.val)
; CHECK-NEXT:    ret i32 0
; CHECK-NEXT: }

; CHECK-LABEL: define internal void @bar(<32 x half> %.0.val) #0 {
; CHECK-NEXT:    ret void
; CHECK-NEXT: }

; CHECK: attributes #0 = { uwtable "min-legal-vector-width"="512" }

define i32 @foo() #0 {
    call void @bar(ptr undef)
    ret i32 0
}

define internal void @bar(ptr) #0 {
    %2 = load <32 x half>, ptr %0, align 4
    ret void
}

```

attributes #0 = { uwtable "min-legal-vector-width"="0" }

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the

editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition,

"submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.



To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

#### ---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

=====  
Software from third parties included in the LLVM Project:

=====  
The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or  
`LICENSE` file at the top containing the specific  
license and restrictions  
which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every  
file.

=====

Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):

=====

The libc++ library is dual licensed under both the University of Illinois  
"BSD-Like" license and the MIT license. As a user of this code you may choose  
to use it under either license. As a contributor, you agree to allow your code  
to be used under both.

Full text of the relevant licenses is included below.

=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2009-2019 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

<http://llvm.org>

Permission

is hereby granted, free of charge, to any person obtaining a copy of  
this software and associated documentation files (the "Software"), to deal with  
the Software without restriction, including without limitation the rights to  
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies  
of the Software, and to permit persons to whom the Software is furnished to do  
so, subject to the following conditions:

\* Redistributions of source code must retain the above copyright notice,  
this list of conditions and the following disclaimers.

\* Redistributions in binary form must reproduce the above copyright notice,  
this list of conditions and the following disclaimers in the

documentation and/or other materials provided with the distribution.

\* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====

Copyright (c) 2009-2014 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all

copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This work is released into the public domain with CC0 1.0. Alternatively, it is licensed under the Apache License 2.0.

-----  
Creative Commons Legal Code

CC0 1.0 Universal

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS DOCUMENT DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED HEREUNDER, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED HEREUNDER.

Statement of Purpose

The laws of most jurisdictions throughout the world automatically confer exclusive Copyright and Related Rights (defined below) upon the creator and subsequent owner(s) (each and all, an "owner") of an original work of authorship and/or a database (each, a "Work").

Certain owners wish to permanently relinquish those rights to a Work for the purpose of contributing to a commons of creative, cultural and scientific works ("Commons") that the public can reliably and without fear of later claims of infringement build upon, modify, incorporate in other works, reuse and redistribute as freely as possible in any form whatsoever and for any purposes, including without limitation commercial purposes. These owners may contribute to the Commons to promote the ideal of a free culture and the further production of creative, cultural and scientific works, or to gain reputation or greater distribution for their Work in part through the use and efforts of others.

For these and/or other purposes and motivations, and without any

expectation of additional consideration or compensation, the person associating CC0 with a Work (the "Affirmer"), to the extent that he or she is an owner of Copyright and Related Rights in the Work, voluntarily elects to apply CC0 to the Work and publicly distribute the Work under its terms, with knowledge of his or her Copyright and Related Rights in the Work and the meaning and intended legal effect of CC0 on those rights.

1. Copyright and Related Rights. A Work made available under CC0 may be protected by copyright and related or neighboring rights ("Copyright and Related Rights"). Copyright and Related Rights include, but are not limited to, the following:

- i. the right to reproduce, adapt, distribute, perform, display, communicate, and translate a Work;
- ii. moral rights retained by the original author(s) and/or performer(s);
- iii. publicity and privacy rights pertaining to a person's image or likeness depicted in a Work;
- iv. rights protecting against unfair competition in regards to a Work, subject to the limitations in paragraph 4(a), below;
- v. rights protecting the extraction, dissemination, use and reuse of data in a Work;
- vi. database rights (such as those arising under Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, and under any national implementation thereof, including any amended or successor version of such directive); and
- vii. other similar, equivalent or corresponding rights throughout the world based on applicable law or treaty, and any national implementations thereof.

2. Waiver. To the greatest extent permitted by, but not in contravention of, applicable law, Affirmer hereby overtly, fully, permanently, irrevocably and unconditionally waives, abandons, and surrenders all of Affirmer's Copyright and Related Rights and associated claims and causes of action, whether now known or unknown (including existing as well as future claims and causes of action), in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "Waiver"). Affirmer makes the Waiver for the benefit of each member of the public at large and to the detriment of Affirmer's heirs and successors, fully intending that such Waiver shall not be subject to revocation, rescission, cancellation, termination, or any other legal or equitable action to disrupt the quiet enjoyment of the Work by the public

as contemplated by Affirmer's express Statement of Purpose.

3. Public License Fallback. Should any part of the Waiver for any reason be judged legally invalid or ineffective under applicable law, then the Waiver shall be preserved to the maximum extent permitted taking into account Affirmer's express Statement of Purpose. In addition, to the extent the Waiver is so judged Affirmer hereby grants to each affected person a royalty-free, non transferable, non sublicensable, non exclusive, irrevocable and unconditional license to exercise Affirmer's Copyright and Related Rights in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "License"). The License shall be deemed effective as of the date CC0 was applied by Affirmer to the Work. Should any part of the License for any reason be judged legally invalid or ineffective under applicable law, such partial invalidity or ineffectiveness shall not invalidate the remainder of the License, and in such case Affirmer hereby affirms that he or she will not (i) exercise any of his or her remaining Copyright and Related Rights in the Work or (ii) assert any associated claims and causes of action with respect to the Work, in either case contrary to Affirmer's express Statement of Purpose.

#### 4. Limitations and Disclaimers.

- a. No trademark or patent rights held by Affirmer are waived, abandoned, surrendered, licensed or otherwise affected by this document.
- b. Affirmer offers the Work as-is and makes no representations or warranties of any kind concerning the Work, express, implied, statutory or otherwise, including without limitation warranties of title, merchantability, fitness for a particular purpose, non infringement, or the absence of latent or other defects, accuracy, or the present or absence of errors, whether or not discoverable, all to the greatest extent permissible under applicable law.
- c. Affirmer disclaims responsibility for clearing rights of other persons that may apply to the Work or any use thereof, including without limitation any person's Copyright and Related Rights in the Work. Further, Affirmer disclaims responsibility for obtaining any necessary consents, permissions or other rights required for any use of the Work.
- d. Affirmer understands and acknowledges that Creative Commons is not a party to this document and has no duty or obligation with respect to this CC0 or use of the Work.

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,

the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity

authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit)

alleging that the Work

or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without



modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2019 Jack O'Connor and Samuel Neves

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -mattr=+m -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefixes=CHECK,RV64I
; RUN: llc -mtriple=riscv64 -mattr=+m,+zba -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-
prefixes=CHECK,RV64ZBA,RV64ZBANOZBB
; RUN: llc -mtriple=riscv64 -mattr=+m,+zba,+zbb -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefixes=CHECK,RV64ZBA,RV64ZBAZBB
```

```
define i64 @slliuv(i64 %a) nounwind {
; RV64I-LABEL: slliuv:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 31
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: slliuv:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  slli.uw a0, a0, 1
; RV64ZBA-NEXT:  ret
%conv1 = shl i64 %a, 1
%shl = and i64 %conv1, 8589934590
ret i64 %shl
}
```

```
define i128 @slliuv_2(i32 signext %0, ptr %1) {
; RV64I-LABEL:
```

```

slliuw_2:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 28
; RV64I-NEXT:  add a1, a1, a0
; RV64I-NEXT:  ld a0, 0(a1)
; RV64I-NEXT:  ld a1, 8(a1)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: slliuw_2:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  slli.uw a0, a0, 4
; RV64ZBA-NEXT:  add a1, a1, a0
; RV64ZBA-NEXT:  ld a0, 0(a1)
; RV64ZBA-NEXT:  ld a1, 8(a1)
; RV64ZBA-NEXT:  ret
%3 = zext i32 %0 to i64
%4 = getelementptr inbounds i128, ptr %1, i64 %3
%5 = load i128, ptr %4
ret i128 %5
}

```

```

define i128 @slliuw_3(i32 signext %0, ptr %1) {
; RV64I-LABEL: slliuw_3:
; RV64I:    # %bb.0:
; RV64I-NEXT:  addi a0, a0, 1
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 28
; RV64I-NEXT:  add a1, a1, a0
; RV64I-NEXT:  ld a0, 0(a1)
; RV64I-NEXT:  ld a1, 8(a1)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: slliuw_3:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  addi a0, a0, 1
; RV64ZBA-NEXT:  slli.uw a0, a0, 4
; RV64ZBA-NEXT:
    add a1, a1, a0
; RV64ZBA-NEXT:  ld a0, 0(a1)
; RV64ZBA-NEXT:  ld a1, 8(a1)
; RV64ZBA-NEXT:  ret
%add = add i32 %0, 1
%3 = zext i32 %add to i64
%4 = getelementptr inbounds i128, ptr %1, i64 %3
%5 = load i128, ptr %4
ret i128 %5
}

```

```

define i64 @adduw(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: adduw:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  srli a1, a1, 32
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: adduw:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  add.uw a0, a1, a0
; RV64ZBA-NEXT:  ret
%and = and i64 %b, 4294967295
%add = add i64 %and, %a
ret i64 %add
}

define signext i8 @adduw_2(i32 signext %0, ptr %1) {
; RV64I-LABEL: adduw_2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  lb a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: adduw_2:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  add.uw a0, a0, a1
; RV64ZBA-NEXT:
    lb a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = zext i32 %0 to i64
%4 = getelementptr inbounds i8, ptr %1, i64 %3
%5 = load i8, ptr %4
ret i8 %5
}

define signext i8 @adduw_3(i32 signext %0, ptr %1) {
; RV64I-LABEL: adduw_3:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi a0, a0, 1
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  lb a0, 0(a0)
; RV64I-NEXT:  ret

```

```

;
; RV64ZBA-LABEL: adduw_3:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  addi a0, a0, 1
; RV64ZBA-NEXT:  add.uw a0, a0, a1
; RV64ZBA-NEXT:  lb a0, 0(a0)
; RV64ZBA-NEXT:  ret
%add = add i32 %0, 1
%3 = zext i32 %add to i64
%4 = getelementptr inbounds i8, ptr %1, i64 %3
%5 = load i8, ptr %4
ret i8 %5
}

```

```

define i64 @zextw_i64(i64 %a) nounwind {
; RV64I-LABEL: zextw_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: zextw_i64:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:
    zext.w a0, a0
; RV64ZBA-NEXT:  ret
%and = and i64 %a, 4294967295
ret i64 %and
}

```

; This makes sure targetShrinkDemandedConstant changes the and immediate to  
; allow zext.w or slli+srli.

```

define i64 @zextw_demandedbits_i64(i64 %0) {
; RV64I-LABEL: zextw_demandedbits_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  ori a0, a0, 1
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: zextw_demandedbits_i64:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  ori a0, a0, 1
; RV64ZBA-NEXT:  zext.w a0, a0
; RV64ZBA-NEXT:  ret
%2 = and i64 %0, 4294967294
%3 = or i64 %2, 1
ret i64 %3
}

```

```
}
```

```
define signext i16 @sh1add(i64 %0, ptr %1) {  
; RV64I-LABEL: sh1add:  
; RV64I:    # %bb.0:  
; RV64I-NEXT:  slli a0, a0, 1  
; RV64I-NEXT:  add a0, a1, a0  
; RV64I-NEXT:  lh a0, 0(a0)  
; RV64I-NEXT:  ret  
;  
; RV64ZBA-LABEL: sh1add:  
; RV64ZBA:    # %bb.0:  
; RV64ZBA-NEXT:  sh1add a0, a0, a1  
; RV64ZBA-NEXT:  lh a0, 0(a0)  
; RV64ZBA-NEXT:  
    ret  
%3 = getelementptr inbounds i16, ptr %1, i64 %0  
%4 = load i16, ptr %3  
ret i16 %4  
}
```

```
define signext i32 @sh2add(i64 %0, ptr %1) {  
; RV64I-LABEL: sh2add:  
; RV64I:    # %bb.0:  
; RV64I-NEXT:  slli a0, a0, 2  
; RV64I-NEXT:  add a0, a1, a0  
; RV64I-NEXT:  lw a0, 0(a0)  
; RV64I-NEXT:  ret  
;  
; RV64ZBA-LABEL: sh2add:  
; RV64ZBA:    # %bb.0:  
; RV64ZBA-NEXT:  sh2add a0, a0, a1  
; RV64ZBA-NEXT:  lw a0, 0(a0)  
; RV64ZBA-NEXT:  ret  
%3 = getelementptr inbounds i32, ptr %1, i64 %0  
%4 = load i32, ptr %3  
ret i32 %4  
}
```

```
define i64 @sh3add(i64 %0, ptr %1) {  
; RV64I-LABEL: sh3add:  
; RV64I:    # %bb.0:  
; RV64I-NEXT:  slli a0, a0, 3  
; RV64I-NEXT:  add a0, a1, a0  
; RV64I-NEXT:  ld a0, 0(a0)  
; RV64I-NEXT:  ret  
;  
; RV64ZBA-LABEL: sh3add:
```

```

; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh3add a0, a0, a1
; RV64ZBA-NEXT:  ld a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = getelementptr inbounds i64, ptr %1, i64 %0
%4 = load i64, ptr %3
ret i64 %4
}

```

```

define signext i16 @sh1adduw(i32
signext %0, ptr %1) {
; RV64I-LABEL: sh1adduw:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 31
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  lh a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh1adduw:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh1add.uw a0, a0, a1
; RV64ZBA-NEXT:  lh a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = zext i32 %0 to i64
%4 = getelementptr inbounds i16, ptr %1, i64 %3
%5 = load i16, ptr %4
ret i16 %5
}

```

```

define i64 @sh1adduw_2(i64 %0, i64 %1) {
; RV64I-LABEL: sh1adduw_2:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 31
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh1adduw_2:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh1add.uw a0, a0, a1
; RV64ZBA-NEXT:  ret
%3 = shl i64 %0, 1
%4 = and i64 %3, 8589934590
%5 = add i64 %4, %1
ret i64 %5
}

```



```

define signext i32 @sh2adduw(i32 signext %0, ptr %1) {
; RV64I-LABEL: sh2adduw:
; RV64I:
    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 30
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  lw a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh2adduw:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh2add.uw a0, a0, a1
; RV64ZBA-NEXT:  lw a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = zext i32 %0 to i64
%4 = getelementptr inbounds i32, ptr %1, i64 %3
%5 = load i32, ptr %4
ret i32 %5
}

```

```

define i64 @sh2adduw_2(i64 %0, i64 %1) {
; RV64I-LABEL: sh2adduw_2:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 30
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh2adduw_2:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh2add.uw a0, a0, a1
; RV64ZBA-NEXT:  ret
%3 = shl i64 %0, 2
%4 = and i64 %3, 17179869180
%5 = add i64 %4, %1
ret i64 %5
}

```

```

define i64 @sh3adduw(i32 signext %0, ptr %1) {
; RV64I-LABEL: sh3adduw:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:
    srli a0, a0, 29
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  ld a0, 0(a0)
; RV64I-NEXT:  ret

```

```

;
; RV64ZBA-LABEL: sh3adduw:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh3add.uw a0, a0, a1
; RV64ZBA-NEXT:  ld a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = zext i32 %0 to i64
%4 = getelementptr inbounds i64, ptr %1, i64 %3
%5 = load i64, ptr %4
ret i64 %5
}

```

```

define i64 @sh3adduw_2(i64 %0, i64 %1) {
; RV64I-LABEL: sh3adduw_2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 29
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh3adduw_2:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh3add.uw a0, a0, a1
; RV64ZBA-NEXT:  ret
%3 = shl i64 %0, 3
%4 = and i64 %3, 34359738360
%5 = add i64 %4, %1
ret i64 %5
}

```

; Type legalization inserts a sext\_inreg after the first add. That add will be  
; selected as sh2add which does not sign extend. SimplifyDandedBits is unable  
; to remove the sext\_inreg because it has multiple uses.

The ashr will use the

; sext\_inreg to become sraiw. This leaves the sext\_inreg only used by the shl.

; If the shl is selected as sllw, we don't need the sext\_inreg.

```
define i64 @sh2add_extra_sext(i32 %x, i32 %y, i32 %z) {
```

```

; RV64I-LABEL: sh2add_extra_sext:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 2
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  sllw a1, a2, a0
; RV64I-NEXT:  sraiw a0, a0, 2
; RV64I-NEXT:  mul a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh2add_extra_sext:
; RV64ZBA:      # %bb.0:

```

```

; RV64ZBA-NEXT:  sh2add a0, a0, a1
; RV64ZBA-NEXT:  sllw a1, a2, a0
; RV64ZBA-NEXT:  sraiw a0, a0, 2
; RV64ZBA-NEXT:  mul a0, a1, a0
; RV64ZBA-NEXT:  ret
%a = shl i32 %x, 2
%b = add i32 %a, %y
%c = shl i32 %z, %b
%d = ashr i32 %b, 2
%e = sext i32 %c to i64
%f = sext i32 %d to i64
%g = mul i64 %e, %f
ret i64 %g
}

```

```

define i64 @addmul6(i64 %a, i64 %b) {
; RV64I-LABEL: addmul6:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a2, 6
; RV64I-NEXT:  mul
a0, a0, a2
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addmul6:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh1add a0, a0, a0
; RV64ZBA-NEXT:  sh1add a0, a0, a1
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 6
%d = add i64 %c, %b
ret i64 %d
}

```

```

define i64 @addmul10(i64 %a, i64 %b) {
; RV64I-LABEL: addmul10:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a2, 10
; RV64I-NEXT:  mul a0, a0, a2
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addmul10:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh2add a0, a0, a0
; RV64ZBA-NEXT:  sh1add a0, a0, a1
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 10

```

```

%d = add i64 %c, %b
ret i64 %d
}

define i64 @addmul12(i64 %a, i64 %b) {
; RV64I-LABEL: addmul12:
; RV64I:      # %bb.0:
; RV64I-NEXT: li a2, 12
; RV64I-NEXT: mul a0, a0, a2
; RV64I-NEXT: add a0, a0, a1
; RV64I-NEXT: ret
;
; RV64ZBA-LABEL: addmul12:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT: sh1add a0, a0, a0
; RV64ZBA-NEXT: sh2add
a0, a0, a1
; RV64ZBA-NEXT: ret
%c = mul i64 %a, 12
%d = add i64 %c, %b
ret i64 %d
}

define i64 @addmul18(i64 %a, i64 %b) {
; RV64I-LABEL: addmul18:
; RV64I:      # %bb.0:
; RV64I-NEXT: li a2, 18
; RV64I-NEXT: mul a0, a0, a2
; RV64I-NEXT: add a0, a0, a1
; RV64I-NEXT: ret
;
; RV64ZBA-LABEL: addmul18:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT: sh3add a0, a0, a0
; RV64ZBA-NEXT: sh1add a0, a0, a1
; RV64ZBA-NEXT: ret
%c = mul i64 %a, 18
%d = add i64 %c, %b
ret i64 %d
}

define i64 @addmul20(i64 %a, i64 %b) {
; RV64I-LABEL: addmul20:
; RV64I:      # %bb.0:
; RV64I-NEXT: li a2, 20
; RV64I-NEXT: mul a0, a0, a2
; RV64I-NEXT: add a0, a0, a1
; RV64I-NEXT: ret

```

```

;
; RV64ZBA-LABEL: addmul20:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh2add a0, a0, a0
; RV64ZBA-NEXT:  sh2add a0, a0, a1
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 20
%d = add i64 %c, %b
ret i64 %d
}

define i64 @addmul24(i64 %a, i64 %b) {
; RV64I-LABEL: addmul24:
; RV64I:
;   # %bb.0:
; RV64I-NEXT:  li a2, 24
; RV64I-NEXT:  mul a0, a0, a2
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addmul24:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh1add a0, a0, a0
; RV64ZBA-NEXT:  sh3add a0, a0, a1
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 24
%d = add i64 %c, %b
ret i64 %d
}

define i64 @addmul36(i64 %a, i64 %b) {
; RV64I-LABEL: addmul36:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a2, 36
; RV64I-NEXT:  mul a0, a0, a2
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addmul36:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh3add a0, a0, a0
; RV64ZBA-NEXT:  sh2add a0, a0, a1
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 36
%d = add i64 %c, %b
ret i64 %d
}

```

```

define i64 @addmul40(i64 %a, i64 %b) {
; RV64I-LABEL: addmul40:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a2, 40
; RV64I-NEXT:  mul a0, a0, a2
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addmul40:
; RV64ZBA:    # %bb.0:
;
; RV64ZBA-NEXT: sh2add a0, a0, a0
; RV64ZBA-NEXT: sh3add a0, a0, a1
; RV64ZBA-NEXT: ret
%c = mul i64 %a, 40
%d = add i64 %c, %b
ret i64 %d
}

```

```

define i64 @addmul72(i64 %a, i64 %b) {
; RV64I-LABEL: addmul72:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a2, 72
; RV64I-NEXT:  mul a0, a0, a2
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addmul72:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT: sh3add a0, a0, a0
; RV64ZBA-NEXT: sh3add a0, a0, a1
; RV64ZBA-NEXT: ret
%c = mul i64 %a, 72
%d = add i64 %c, %b
ret i64 %d
}

```

```

define i64 @mul96(i64 %a) {
; RV64I-LABEL: mul96:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 96
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul96:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT: sh1add a0, a0, a0
; RV64ZBA-NEXT: slli a0, a0, 5

```

```

; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 96
ret i64 %c
}

define i64 @mul160(i64 %a) {
; RV64I-LABEL: mul160:
; RV64I:      # %bb.0:
; RV64I-NEXT:
    li a1, 160
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul160:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh2add a0, a0, a0
; RV64ZBA-NEXT:  slli a0, a0, 5
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 160
ret i64 %c
}

define i64 @mul288(i64 %a) {
; RV64I-LABEL: mul288:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 288
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul288:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh3add a0, a0, a0
; RV64ZBA-NEXT:  slli a0, a0, 5
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 288
ret i64 %c
}

define i64 @zext_mul96(i32 signext %a) {
; RV64I-LABEL: zext_mul96:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 3
; RV64I-NEXT:  slli a1, a1, 37
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  mulhu a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: zext_mul96:

```

```

; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  slli.uw a0, a0, 5
; RV64ZBA-NEXT:  sh1add a0, a0, a0
; RV64ZBA-NEXT:  ret
%b = zext
i32 %a to i64
%c = mul i64 %b, 96
ret i64 %c
}

define i64 @zext_mul160(i32 signext %a) {
; RV64I-LABEL: zext_mul160:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a1, 5
; RV64I-NEXT:  slli a1, a1, 37
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  mulhu a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: zext_mul160:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  slli.uw a0, a0, 5
; RV64ZBA-NEXT:  sh2add a0, a0, a0
; RV64ZBA-NEXT:  ret
%b = zext i32 %a to i64
%c = mul i64 %b, 160
ret i64 %c
}

define i64 @zext_mul288(i32 signext %a) {
; RV64I-LABEL: zext_mul288:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a1, 9
; RV64I-NEXT:  slli a1, a1, 37
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  mulhu a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: zext_mul288:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  slli.uw a0, a0, 5
; RV64ZBA-NEXT:  sh3add a0, a0, a0
; RV64ZBA-NEXT:  ret
%b = zext i32 %a to i64
%c = mul i64 %b, 288
ret i64 %c
}

```



```

; We can't use slli.uw becaues
the shift amount is more than 31.
; FIXME: The zext.w is unneeded.
define i64 @zext_mul12884901888(i32 signext %a) {
; RV64I-LABEL: zext_mul12884901888:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  li a1, 3
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: zext_mul12884901888:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  andi a0, a0, -1
; RV64ZBA-NEXT:  sh1add a0, a0, a0
; RV64ZBA-NEXT:  slli a0, a0, 32
; RV64ZBA-NEXT:  ret
%b = zext i32 %a to i64
%c = mul i64 %b, 12884901888
ret i64 %c
}

; We can't use slli.uw becaues the shift amount is more than 31.
; FIXME: The zext.w is unneeded.
define i64 @zext_mul21474836480(i32 signext %a) {
; RV64I-LABEL: zext_mul21474836480:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  li a1, 5
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:
ret
;
; RV64ZBA-LABEL: zext_mul21474836480:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  andi a0, a0, -1
; RV64ZBA-NEXT:  sh2add a0, a0, a0
; RV64ZBA-NEXT:  slli a0, a0, 32
; RV64ZBA-NEXT:  ret
%b = zext i32 %a to i64
%c = mul i64 %b, 21474836480
ret i64 %c
}

```

; We can't use slli.uw because the shift amount is more than 31.

; FIXME: The zext.w is unneeded.

define i64 @zext\_mul38654705664(i32 signext %a) {

; RV64I-LABEL: zext\_mul38654705664:

; RV64I: # %bb.0:

; RV64I-NEXT: slli a0, a0, 32

; RV64I-NEXT: srli a0, a0, 32

; RV64I-NEXT: li a1, 9

; RV64I-NEXT: slli a1, a1, 32

; RV64I-NEXT: mul a0, a0, a1

; RV64I-NEXT: ret

;

; RV64ZBA-LABEL: zext\_mul38654705664:

; RV64ZBA: # %bb.0:

; RV64ZBA-NEXT: andi a0, a0, -1

; RV64ZBA-NEXT: sh3add a0, a0, a0

; RV64ZBA-NEXT: slli a0, a0, 32

; RV64ZBA-NEXT: ret

%b = zext i32 %a to i64

%c = mul i64 %b, 38654705664

ret i64 %c

}

define i64 @sh1add\_imm(i64 %0) {

; CHECK-LABEL: sh1add\_imm:

;

CHECK: # %bb.0:

; CHECK-NEXT: slli a0, a0, 1

; CHECK-NEXT: addi a0, a0, 5

; CHECK-NEXT: ret

%a = shl i64 %0, 1

%b = add i64 %a, 5

ret i64 %b

}

define i64 @sh2add\_imm(i64 %0) {

; CHECK-LABEL: sh2add\_imm:

; CHECK: # %bb.0:

; CHECK-NEXT: slli a0, a0, 2

; CHECK-NEXT: addi a0, a0, -6

; CHECK-NEXT: ret

%a = shl i64 %0, 2

%b = add i64 %a, -6

ret i64 %b

}

define i64 @sh3add\_imm(i64 %0) {

; CHECK-LABEL: sh3add\_imm:

```

; CHECK:    # %bb.0:
; CHECK-NEXT:  slli a0, a0, 3
; CHECK-NEXT:  addi a0, a0, 7
; CHECK-NEXT:  ret
%a = shl i64 %0, 3
%b = add i64 %a, 7
ret i64 %b
}

define i64 @sh1adduw_imm(i32 signext %0) {
; RV64I-LABEL: sh1adduw_imm:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 31
; RV64I-NEXT:  addi a0, a0, 11
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh1adduw_imm:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  slli.uw a0, a0, 1
; RV64ZBA-NEXT:  addi a0, a0, 11
; RV64ZBA-NEXT:  ret
%a =
zext i32 %0 to i64
%b = shl i64 %a, 1
%c = add i64 %b, 11
ret i64 %c
}

define i64 @sh2adduw_imm(i32 signext %0) {
; RV64I-LABEL: sh2adduw_imm:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 30
; RV64I-NEXT:  addi a0, a0, -12
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh2adduw_imm:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  slli.uw a0, a0, 2
; RV64ZBA-NEXT:  addi a0, a0, -12
; RV64ZBA-NEXT:  ret
%a = zext i32 %0 to i64
%b = shl i64 %a, 2
%c = add i64 %b, -12
ret i64 %c
}

```

```

define i64 @sh3adduw_imm(i32 signext %0) {
; RV64I-LABEL: sh3adduw_imm:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 29
; RV64I-NEXT:  addi a0, a0, 13
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh3adduw_imm:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  slli.uw a0, a0, 3
; RV64ZBA-NEXT:  addi a0, a0, 13
; RV64ZBA-NEXT:  ret
%a = zext i32 %0 to i64
%b = shl i64 %a, 3
%c = add i64 %b, 13
ret i64 %c
}

```

```

define i64
@adduw_imm(i32 signext %0) nounwind {
; RV64I-LABEL: adduw_imm:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  addi a0, a0, 5
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: adduw_imm:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  zext.w a0, a0
; RV64ZBA-NEXT:  addi a0, a0, 5
; RV64ZBA-NEXT:  ret
%a = zext i32 %0 to i64
%b = add i64 %a, 5
ret i64 %b
}

```

```

define i64 @mul258(i64 %a) {
; CHECK-LABEL: mul258:
; CHECK:      # %bb.0:
; CHECK-NEXT:  li a1, 258
; CHECK-NEXT:  mul a0, a0, a1
; CHECK-NEXT:  ret
%c = mul i64 %a, 258
ret i64 %c
}

```

```

define i64 @mul260(i64 %a) {
; CHECK-LABEL: mul260:
; CHECK:      # %bb.0:
; CHECK-NEXT:  li a1, 260
; CHECK-NEXT:  mul a0, a0, a1
; CHECK-NEXT:  ret
%c = mul i64 %a, 260
ret i64 %c
}

```

```

define i64 @mul264(i64 %a) {
; CHECK-LABEL: mul264:
; CHECK:      # %bb.0:
; CHECK-NEXT:  li a1, 264
; CHECK-NEXT:  mul a0, a0, a1
; CHECK-NEXT:  ret
%c = mul i64 %a, 264
ret i64
%c
}

```

```

define i64 @imm_zextw() nounwind {
; RV64I-LABEL: imm_zextw:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a0, 1
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  addi a0, a0, -2
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: imm_zextw:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  li a0, -2
; RV64ZBA-NEXT:  zext.w a0, a0
; RV64ZBA-NEXT:  ret
ret i64 4294967294 ; -2 in 32 bits.
}

```

```

define i64 @mul11(i64 %a) {
; RV64I-LABEL: mul11:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 11
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul11:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh2add a1, a0, a0
; RV64ZBA-NEXT:  sh1add a0, a1, a0

```

```
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 11
ret i64 %c
}
```

```
define i64 @mul19(i64 %a) {
; RV64I-LABEL: mul19:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 19
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul19:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh3add a1, a0, a0
; RV64ZBA-NEXT:  sh1add a0,
a1, a0
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 19
ret i64 %c
}
```

```
define i64 @mul13(i64 %a) {
; RV64I-LABEL: mul13:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 13
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul13:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh1add a1, a0, a0
; RV64ZBA-NEXT:  sh2add a0, a1, a0
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 13
ret i64 %c
}
```

```
define i64 @mul21(i64 %a) {
; RV64I-LABEL: mul21:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 21
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul21:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh2add a1, a0, a0
```

```
; RV64ZBA-NEXT:  sh2add a0, a1, a0
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 21
ret i64 %c
}
```

```
define i64 @mul37(i64 %a) {
; RV64I-LABEL: mul37:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 37
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul37:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT: sh3add a1, a0, a0
; RV64ZBA-NEXT:
    sh2add a0, a1, a0
; RV64ZBA-NEXT: ret
%c = mul i64 %a, 37
ret i64 %c
}
```

```
define i64 @mul25(i64 %a) {
; RV64I-LABEL: mul25:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 25
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul25:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT: sh1add a1, a0, a0
; RV64ZBA-NEXT: sh3add a0, a1, a0
; RV64ZBA-NEXT: ret
%c = mul i64 %a, 25
ret i64 %c
}
```

```
define i64 @mul41(i64 %a) {
; RV64I-LABEL: mul41:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 41
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul41:
; RV64ZBA:    # %bb.0:
```

```
; RV64ZBA-NEXT: sh2add a1, a0, a0
; RV64ZBA-NEXT: sh3add a0, a1, a0
; RV64ZBA-NEXT: ret
%c = mul i64 %a, 41
ret i64 %c
}
```

```
define i64 @mul73(i64 %a) {
; RV64I-LABEL: mul73:
; RV64I:    # %bb.0:
; RV64I-NEXT: li a1, 73
; RV64I-NEXT: mul a0, a0, a1
; RV64I-NEXT: ret
;
; RV64ZBA-LABEL: mul73:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT: sh3add
a1, a0, a0
; RV64ZBA-NEXT: sh3add a0, a1, a0
; RV64ZBA-NEXT: ret
%c = mul i64 %a, 73
ret i64 %c
}
```

```
define i64 @mul27(i64 %a) {
; RV64I-LABEL: mul27:
; RV64I:    # %bb.0:
; RV64I-NEXT: li a1, 27
; RV64I-NEXT: mul a0, a0, a1
; RV64I-NEXT: ret
;
; RV64ZBA-LABEL: mul27:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT: sh3add a0, a0, a0
; RV64ZBA-NEXT: sh1add a0, a0, a0
; RV64ZBA-NEXT: ret
%c = mul i64 %a, 27
ret i64 %c
}
```

```
define i64 @mul45(i64 %a) {
; RV64I-LABEL: mul45:
; RV64I:    # %bb.0:
; RV64I-NEXT: li a1, 45
; RV64I-NEXT: mul a0, a0, a1
; RV64I-NEXT: ret
;
; RV64ZBA-LABEL: mul45:
```



```

; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh3add a0, a0, a0
; RV64ZBA-NEXT:  sh2add a0, a0, a0
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 45
ret i64 %c
}

```

```

define i64 @mul81(i64 %a) {
; RV64I-LABEL: mul81:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a1, 81
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul81:
; RV64ZBA:    # %bb.0:
;
RV64ZBA-NEXT:  sh3add a0, a0, a0
; RV64ZBA-NEXT:  sh3add a0, a0, a0
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 81
ret i64 %c
}

```

```

define i64 @mul4098(i64 %a) {
; RV64I-LABEL: mul4098:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a1, a0, 1
; RV64I-NEXT:  slli a0, a0, 12
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul4098:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  slli a1, a0, 12
; RV64ZBA-NEXT:  sh1add a0, a0, a1
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 4098
ret i64 %c
}

```

```

define i64 @mul4100(i64 %a) {
; RV64I-LABEL: mul4100:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a1, a0, 2
; RV64I-NEXT:  slli a0, a0, 12
; RV64I-NEXT:  add a0, a0, a1

```

```

; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul4100:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  slli a1, a0, 12
; RV64ZBA-NEXT:  sh2add a0, a0, a1
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 4100
ret i64 %c
}

```

```

define i64 @mul4104(i64 %a) {
; RV64I-LABEL: mul4104:
; RV64I:        # %bb.0:
; RV64I-NEXT:
    slli a1, a0, 3
; RV64I-NEXT:  slli a0, a0, 12
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul4104:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  slli a1, a0, 12
; RV64ZBA-NEXT:  sh3add a0, a0, a1
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 4104
ret i64 %c
}

```

```

define signext i32 @mulw192(i32 signext %a) {
; CHECK-LABEL: mulw192:
; CHECK:        # %bb.0:
; CHECK-NEXT:   li a1, 192
; CHECK-NEXT:   mulw a0, a0, a1
; CHECK-NEXT:   ret
%c = mul i32 %a, 192
ret i32 %c
}

```

```

define signext i32 @mulw320(i32 signext %a) {
; CHECK-LABEL: mulw320:
; CHECK:        # %bb.0:
; CHECK-NEXT:   li a1, 320
; CHECK-NEXT:   mulw a0, a0, a1
; CHECK-NEXT:   ret
%c = mul i32 %a, 320
ret i32 %c
}

```

```

define signext i32 @mulw576(i32 signext %a) {
; CHECK-LABEL: mulw576:
; CHECK:      # %bb.0:
; CHECK-NEXT:  li a1, 576
; CHECK-NEXT:  mulw a0, a0, a1
; CHECK-NEXT:  ret
%c = mul i32 %a, 576
ret i32 %c
}

```

```

define i64 @add4104(i64 %a) {
; RV64I-LABEL: add4104:
; RV64I:
    # %bb.0:
; RV64I-NEXT:  lui a1, 1
; RV64I-NEXT:  addiw a1, a1, 8
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: add4104:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  li a1, 1026
; RV64ZBA-NEXT:  sh2add a0, a1, a0
; RV64ZBA-NEXT:  ret
%c = add i64 %a, 4104
ret i64 %c
}

```

```

define i64 @add8208(i64 %a) {
; RV64I-LABEL: add8208:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a1, 2
; RV64I-NEXT:  addiw a1, a1, 16
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: add8208:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  li a1, 1026
; RV64ZBA-NEXT:  sh3add a0, a1, a0
; RV64ZBA-NEXT:  ret
%c = add i64 %a, 8208
ret i64 %c
}

```

; Make sure we prefer LUI for the 8192 instead of using sh3add.

```

define signext i32 @add8192_i32(i32 signext %a) {

```

```

; CHECK-LABEL: add8192_i32:
; CHECK:      # %bb.0:
; CHECK-NEXT:  lui a1, 2
; CHECK-NEXT:  addw a0, a0, a1
; CHECK-NEXT:  ret
%c = add i32 %a, 8192
ret i32 %c
}

```

```

; Make sure we prefer LUI
for the 8192 instead of using sh3add.
define i64 @add8192(i64 %a) {
; CHECK-LABEL: add8192:
; CHECK:      # %bb.0:
; CHECK-NEXT:  lui a1, 2
; CHECK-NEXT:  add a0, a0, a1
; CHECK-NEXT:  ret
%c = add i64 %a, 8192
ret i64 %c
}

```

```

define signext i32 @addshl32_5_6(i32 signext %a, i32 signext %b) {
; RV64I-LABEL: addshl32_5_6:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 5
; RV64I-NEXT:  slli a1, a1, 6
; RV64I-NEXT:  addw a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addshl32_5_6:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh1add a0, a1, a0
; RV64ZBA-NEXT:  slliw a0, a0, 5
; RV64ZBA-NEXT:  ret
%c = shl i32 %a, 5
%d = shl i32 %b, 6
%e = add i32 %c, %d
ret i32 %e
}

```

```

define i64 @addshl64_5_6(i64 %a, i64 %b) {
; RV64I-LABEL: addshl64_5_6:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 5
; RV64I-NEXT:  slli a1, a1, 6
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;

```

```

; RV64ZBA-LABEL: addshl64_5_6:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:
    sh1add a0, a1, a0
; RV64ZBA-NEXT:  slli a0, a0, 5
; RV64ZBA-NEXT:  ret
    %c = shl i64 %a, 5
    %d = shl i64 %b, 6
    %e = add i64 %c, %d
    ret i64 %e
}

```

```

define signext i32 @addshl32_5_7(i32 signext %a, i32 signext %b) {
; RV64I-LABEL: addshl32_5_7:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 5
; RV64I-NEXT:  slli a1, a1, 7
; RV64I-NEXT:  addw a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addshl32_5_7:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh2add a0, a1, a0
; RV64ZBA-NEXT:  slliw a0, a0, 5
; RV64ZBA-NEXT:  ret
    %c = shl i32 %a, 5
    %d = shl i32 %b, 7
    %e = add i32 %c, %d
    ret i32 %e
}

```

```

define i64 @addshl64_5_7(i64 %a, i64 %b) {
; RV64I-LABEL: addshl64_5_7:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 5
; RV64I-NEXT:  slli a1, a1, 7
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addshl64_5_7:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh2add a0, a1, a0
; RV64ZBA-NEXT:  slli a0, a0, 5
; RV64ZBA-NEXT:  ret
    %c
    = shl i64 %a, 5
    %d = shl i64 %b, 7
    %e = add i64 %c, %d

```

```

ret i64 %e
}

define signext i32 @addshl32_5_8(i32 signext %a, i32 signext %b) {
; RV64I-LABEL: addshl32_5_8:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 5
; RV64I-NEXT:  slli a1, a1, 8
; RV64I-NEXT:  addw a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addshl32_5_8:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh3add a0, a1, a0
; RV64ZBA-NEXT:  slliw a0, a0, 5
; RV64ZBA-NEXT:  ret
%c = shl i32 %a, 5
%d = shl i32 %b, 8
%e = add i32 %c, %d
ret i32 %e
}

define i64 @addshl64_5_8(i64 %a, i64 %b) {
; RV64I-LABEL: addshl64_5_8:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 5
; RV64I-NEXT:  slli a1, a1, 8
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addshl64_5_8:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh3add a0, a1, a0
; RV64ZBA-NEXT:  slli a0, a0, 5
; RV64ZBA-NEXT:  ret
%c = shl i64 %a, 5
%d = shl i64 %b, 8
%e = add i64 %c, %d
ret i64 %e
}

; Make
sure we use sext.h+slli+srli for Zba+Zbb.
; FIXME: The RV64I and Zba only cases can be done with only 3 shifts.
define zeroext i32 @sext_ashr_zext_i8(i8 %a) nounwind {
; RV64I-LABEL: sext_ashr_zext_i8:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 24

```

```

; RV64I-NEXT:  sraiw a0, a0, 31
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  ret
;
; RV64ZBANNOZBB-LABEL: sext_ashr_zext_i8:
; RV64ZBANNOZBB:    # %bb.0:
; RV64ZBANNOZBB-NEXT:  slli a0, a0, 24
; RV64ZBANNOZBB-NEXT:  sraiw a0, a0, 31
; RV64ZBANNOZBB-NEXT:  zext.w a0, a0
; RV64ZBANNOZBB-NEXT:  ret
;
; RV64ZBAZBB-LABEL: sext_ashr_zext_i8:
; RV64ZBAZBB:    # %bb.0:
; RV64ZBAZBB-NEXT:  sext.b a0, a0
; RV64ZBAZBB-NEXT:  sraiw a0, a0, 9
; RV64ZBAZBB-NEXT:  zext.w a0, a0
; RV64ZBAZBB-NEXT:  ret
%ext = sext i8 %a to i32
%1 = ashr i32 %ext, 9
ret i32 %1
}

; Make sure we use sext.h+slli+srli for Zba+Zbb.
; FIXME: The RV64I and Zba only cases can be done with only 3 shifts.
define
zeroext i32 @sext_ashr_zext_i16(i16 %a) nounwind {
; RV64I-LABEL: sext_ashr_zext_i16:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 16
; RV64I-NEXT:  sraiw a0, a0, 25
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  ret
;
; RV64ZBANNOZBB-LABEL: sext_ashr_zext_i16:
; RV64ZBANNOZBB:    # %bb.0:
; RV64ZBANNOZBB-NEXT:  slli a0, a0, 16
; RV64ZBANNOZBB-NEXT:  sraiw a0, a0, 25
; RV64ZBANNOZBB-NEXT:  zext.w a0, a0
; RV64ZBANNOZBB-NEXT:  ret
;
; RV64ZBAZBB-LABEL: sext_ashr_zext_i16:
; RV64ZBAZBB:    # %bb.0:
; RV64ZBAZBB-NEXT:  slli a0, a0, 48
; RV64ZBAZBB-NEXT:  srai a0, a0, 57
; RV64ZBAZBB-NEXT:  zext.w a0, a0
; RV64ZBAZBB-NEXT:  ret

```

```

%ext = sext i16 %a to i32
%1 = ashr i32 %ext, 9
ret i32 %1
}

```

; This the IR you get from InstCombine if take the difference of 2 pointers and  
; cast is to unsigned before using as an index.

```

define signext i16 @sh1adduw_ptrdiff(i64 %diff, ptr %baseptr) {
; RV64I-LABEL: sh1adduw_ptrdiff:
; RV64I:
;   # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  slli a2, a2, 33
; RV64I-NEXT:  addi a2, a2, -2
; RV64I-NEXT:  and a0, a0, a2
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  lh a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh1adduw_ptrdiff:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  srli a0, a0, 1
; RV64ZBA-NEXT:  sh1add.uw a0, a0, a1
; RV64ZBA-NEXT:  lh a0, 0(a0)
; RV64ZBA-NEXT:  ret
%ptrdiff = lshr exact i64 %diff, 1
%cast = and i64 %ptrdiff, 4294967295
%ptr = getelementptr inbounds i16, ptr %baseptr, i64 %cast
%res = load i16, ptr %ptr
ret i16 %res
}

```

```

define signext i32 @sh2adduw_ptrdiff(i64 %diff, ptr %baseptr) {
; RV64I-LABEL: sh2adduw_ptrdiff:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  slli a2, a2, 34
; RV64I-NEXT:  addi a2, a2, -4
; RV64I-NEXT:  and a0, a0, a2
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  lw a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh2adduw_ptrdiff:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:
;   srli a0, a0, 2
; RV64ZBA-NEXT:  sh2add.uw a0, a0, a1

```



```

; RV64ZBA-NEXT:  lw a0, 0(a0)
; RV64ZBA-NEXT:  ret
%ptrdiff = lshr exact i64 %diff, 2
%cast = and i64 %ptrdiff, 4294967295
%ptr = getelementptr inbounds i32, ptr %baseptr, i64 %cast
%res = load i32, ptr %ptr
ret i32 %res
}

```

```

define i64 @sh3adduw_ptrdiff(i64 %diff, ptr %baseptr) {
; RV64I-LABEL: sh3adduw_ptrdiff:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  slli a2, a2, 35
; RV64I-NEXT:  addi a2, a2, -8
; RV64I-NEXT:  and a0, a0, a2
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  ld a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh3adduw_ptrdiff:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  srli a0, a0, 3
; RV64ZBA-NEXT:  sh3add.uw a0, a0, a1
; RV64ZBA-NEXT:  ld a0, 0(a0)
; RV64ZBA-NEXT:  ret
%ptrdiff = lshr exact i64 %diff, 3
%cast = and i64 %ptrdiff, 4294967295
%ptr = getelementptr inbounds i64, ptr %baseptr, i64 %cast
%res = load i64, ptr %ptr
ret i64 %res
}

```

```

define signext
i16 @srliw_1_sh1add(ptr %0, i32 signext %1) {
; RV64I-LABEL: srliw_1_sh1add:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srliw a1, a1, 1
; RV64I-NEXT:  slli a1, a1, 1
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  lh a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: srliw_1_sh1add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  srliw a1, a1, 1
; RV64ZBA-NEXT:  sh1add a0, a1, a0
; RV64ZBA-NEXT:  lh a0, 0(a0)

```

```

; RV64ZBA-NEXT:  ret
%3 = lshr i32 %1, 1
%4 = zext i32 %3 to i64
%5 = getelementptr inbounds i16, ptr %0, i64 %4
%6 = load i16, ptr %5, align 2
ret i16 %6
}

define i128 @slliuw_ptrdiff(i64 %diff, ptr %baseptr) {
; RV64I-LABEL: slliuw_ptrdiff:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  slli a2, a2, 36
; RV64I-NEXT:  addi a2, a2, -16
; RV64I-NEXT:  and a0, a0, a2
; RV64I-NEXT:  add a1, a1, a0
; RV64I-NEXT:  ld a0, 0(a1)
; RV64I-NEXT:  ld a1, 8(a1)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: slliuw_ptrdiff:
; RV64ZBA:      # %bb.0:
;
RV64ZBA-NEXT:  srli a0, a0, 4
; RV64ZBA-NEXT:  slli.uw a0, a0, 4
; RV64ZBA-NEXT:  add a1, a1, a0
; RV64ZBA-NEXT:  ld a0, 0(a1)
; RV64ZBA-NEXT:  ld a1, 8(a1)
; RV64ZBA-NEXT:  ret
%ptrdiff = lshr exact i64 %diff, 4
%cast = and i64 %ptrdiff, 4294967295
%ptr = getelementptr inbounds i128, ptr %baseptr, i64 %cast
%res = load i128, ptr %ptr
ret i128 %res
}

define signext i32 @srliw_2_sh2add(ptr %0, i32 signext %1) {
; RV64I-LABEL: srliw_2_sh2add:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srliw a1, a1, 2
; RV64I-NEXT:  slli a1, a1, 2
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  lw a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: srliw_2_sh2add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  srliw a1, a1, 2

```

```

; RV64ZBA-NEXT:  sh2add a0, a1, a0
; RV64ZBA-NEXT:  lw a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = lshr i32 %1, 2
%4 = zext i32 %3 to i64
%5 = getelementptr inbounds i32, ptr %0, i64 %4
%6 = load i32, ptr %5, align 4
ret i32 %6
}

```

```

define i64 @srliw_3_sh3add(ptr
%0, i32 signext %1) {
; RV64I-LABEL: srliw_3_sh3add:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srliw a1, a1, 3
; RV64I-NEXT:  slli a1, a1, 3
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ld a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: srliw_3_sh3add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  srliw a1, a1, 3
; RV64ZBA-NEXT:  sh3add a0, a1, a0
; RV64ZBA-NEXT:  ld a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = lshr i32 %1, 3
%4 = zext i32 %3 to i64
%5 = getelementptr inbounds i64, ptr %0, i64 %4
%6 = load i64, ptr %5, align 8
ret i64 %6
}

```

```

define signext i32 @srliw_1_sh2add(ptr %0, i32 signext %1) {
; RV64I-LABEL: srliw_1_sh2add:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srliw a1, a1, 1
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  srli a1, a1, 30
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  lw a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: srliw_1_sh2add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  srliw a1, a1, 1
; RV64ZBA-NEXT:  sh2add a0, a1, a0
;

```

```

RV64ZBA-NEXT:  lw a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = lshr i32 %1, 1
%4 = zext i32 %3 to i64
%5 = getelementptr inbounds i32, ptr %0, i64 %4
%6 = load i32, ptr %5, align 4
ret i32 %6
}

define i64 @srliw_1_sh3add(ptr %0, i32 signext %1) {
; RV64I-LABEL: srliw_1_sh3add:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srliw a1, a1, 1
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  srli a1, a1, 29
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ld a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: srliw_1_sh3add:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  srliw a1, a1, 1
; RV64ZBA-NEXT:  sh3add a0, a1, a0
; RV64ZBA-NEXT:  ld a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = lshr i32 %1, 1
%4 = zext i32 %3 to i64
%5 = getelementptr inbounds i64, ptr %0, i64 %4
%6 = load i64, ptr %5, align 8
ret i64 %6
}

define i64 @srliw_2_sh3add(ptr %0, i32 signext %1) {
; RV64I-LABEL: srliw_2_sh3add:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srliw a1, a1, 2
; RV64I-NEXT:  slli a1, a1, 32
;
; RV64I-NEXT:  srli a1, a1, 29
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ld a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: srliw_2_sh3add:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  srliw a1, a1, 2
; RV64ZBA-NEXT:  sh3add a0, a1, a0
; RV64ZBA-NEXT:  ld a0, 0(a0)

```

```
; RV64ZBA-NEXT:  ret
%3 = lshr i32 %1, 2
%4 = zext i32 %3 to i64
%5 = getelementptr inbounds i64, ptr %0, i64 %4
%6 = load i64, ptr %5, align 8
ret i64 %6
}
```

```
define signext i16 @srliw_2_sh1add(ptr %0, i32 signext %1) {
; RV64I-LABEL: srliw_2_sh1add:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srliw a1, a1, 2
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  srli a1, a1, 31
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  lh a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: srliw_2_sh1add:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  srliw a1, a1, 2
; RV64ZBA-NEXT:  sh1add a0, a1, a0
; RV64ZBA-NEXT:  lh a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = lshr i32 %1, 2
%4 = zext i32 %3 to i64
%5 = getelementptr
inbounds i16, ptr %0, i64 %4
%6 = load i16, ptr %5, align 2
ret i16 %6
}
```

```
define signext i32 @srliw_3_sh2add(ptr %0, i32 signext %1) {
; RV64I-LABEL: srliw_3_sh2add:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srliw a1, a1, 3
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  srli a1, a1, 30
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  lw a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: srliw_3_sh2add:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  srliw a1, a1, 3
; RV64ZBA-NEXT:  sh2add a0, a1, a0
; RV64ZBA-NEXT:  lw a0, 0(a0)
}
```

```

; RV64ZBA-NEXT:  ret
%3 = lshr i32 %1, 3
%4 = zext i32 %3 to i64
%5 = getelementptr inbounds i32, ptr %0, i64 %4
%6 = load i32, ptr %5, align 4
ret i32 %6
}

define i64 @srliw_4_sh3add(ptr %0, i32 signext %1) {
; RV64I-LABEL: srliw_4_sh3add:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srliw a1, a1, 4
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  srli a1, a1, 29
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ld a0, 0(a0)
; RV64I-NEXT:
    ret
;
; RV64ZBA-LABEL: srliw_4_sh3add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  srliw a1, a1, 4
; RV64ZBA-NEXT:  sh3add a0, a1, a0
; RV64ZBA-NEXT:  ld a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = lshr i32 %1, 4
%4 = zext i32 %3 to i64
%5 = getelementptr inbounds i64, ptr %0, i64 %4
%6 = load i64, ptr %5, align 8
ret i64 %6
}

define signext i32 @srli_1_sh2add(ptr %0, i64 %1) {
; RV64I-LABEL: srli_1_sh2add:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a1, a1, 1
; RV64I-NEXT:  andi a1, a1, -4
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  lw a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: srli_1_sh2add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  srli a1, a1, 1
; RV64ZBA-NEXT:  sh2add a0, a1, a0
; RV64ZBA-NEXT:  lw a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = lshr i64 %1, 1

```

```

%4 = getelementptr inbounds i32, ptr %0, i64 %3
%5 = load i32, ptr %4, align 4
ret i32 %5
}

```

```

define i64 @srli_2_sh3add(ptr %0, i64 %1) {
; RV64I-LABEL: srli_2_sh3add:
; RV64I:      # %bb.0:
;
; RV64I-NEXT:  slli a1, a1, 1
; RV64I-NEXT:  andi a1, a1, -8
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ld a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: srli_2_sh3add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  srli a1, a1, 2
; RV64ZBA-NEXT:  sh3add a0, a1, a0
; RV64ZBA-NEXT:  ld a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = lshr i64 %1, 2
%4 = getelementptr inbounds i64, ptr %0, i64 %3
%5 = load i64, ptr %4, align 8
ret i64 %5
}

```

```

define signext i16 @srli_2_sh1add(ptr %0, i64 %1) {
; RV64I-LABEL: srli_2_sh1add:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srli a1, a1, 1
; RV64I-NEXT:  andi a1, a1, -2
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  lh a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: srli_2_sh1add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  srli a1, a1, 2
; RV64ZBA-NEXT:  sh1add a0, a1, a0
; RV64ZBA-NEXT:  lh a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = lshr i64 %1, 2
%4 = getelementptr inbounds i16, ptr %0, i64 %3
%5 = load i16, ptr %4, align
2
ret i16 %5
}

```

```

define signext i32 @srli_3_sh2add(ptr %0, i64 %1) {
; RV64I-LABEL: srli_3_sh2add:
; RV64I:      # %bb.0:
; RV64I-NEXT: srli a1, a1, 1
; RV64I-NEXT: andi a1, a1, -4
; RV64I-NEXT: add a0, a0, a1
; RV64I-NEXT: lw a0, 0(a0)
; RV64I-NEXT: ret
;
; RV64ZBA-LABEL: srli_3_sh2add:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT: srli a1, a1, 3
; RV64ZBA-NEXT: sh2add a0, a1, a0
; RV64ZBA-NEXT: lw a0, 0(a0)
; RV64ZBA-NEXT: ret
%3 = lshr i64 %1, 3
%4 = getelementptr inbounds i32, ptr %0, i64 %3
%5 = load i32, ptr %4, align 4
ret i32 %5
}

```

```

define i64 @srli_4_sh3add(ptr %0, i64 %1) {
; RV64I-LABEL: srli_4_sh3add:
; RV64I:      # %bb.0:
; RV64I-NEXT: srli a1, a1, 1
; RV64I-NEXT: andi a1, a1, -8
; RV64I-NEXT: add a0, a0, a1
; RV64I-NEXT: ld a0, 0(a0)
; RV64I-NEXT: ret
;
; RV64ZBA-LABEL: srli_4_sh3add:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT: srli a1, a1, 4
; RV64ZBA-NEXT: sh3add a0, a1, a0
; RV64ZBA-NEXT: ld a0, 0(a0)
; RV64ZBA-NEXT:
ret
%3 = lshr i64 %1, 4
%4 = getelementptr inbounds i64, ptr %0, i64 %3
%5 = load i64, ptr %4, align 8
ret i64 %5
}

```

```

define signext i16 @shl_2_sh1add(ptr %0, i32 signext %1) {
; RV64I-LABEL: shl_2_sh1add:
; RV64I:      # %bb.0:
; RV64I-NEXT: slli a1, a1, 2

```



```

; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  srli a1, a1, 31
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  lh a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: shl_2_sh1add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  slli a1, a1, 2
; RV64ZBA-NEXT:  zext.w a1, a1
; RV64ZBA-NEXT:  sh1add a0, a1, a0
; RV64ZBA-NEXT:  lh a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = shl i32 %1, 2
%4 = zext i32 %3 to i64
%5 = getelementptr inbounds i16, ptr %0, i64 %4
%6 = load i16, ptr %5, align 2
ret i16 %6
}

define signext i32 @shl_16_sh2add(ptr %0, i32 signext %1) {
; RV64I-LABEL: shl_16_sh2add:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a1, a1, 16
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  srli a1, a1, 30
;
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  lw a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: shl_16_sh2add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  slli a1, a1, 16
; RV64ZBA-NEXT:  zext.w a1, a1
; RV64ZBA-NEXT:  sh2add a0, a1, a0
; RV64ZBA-NEXT:  lw a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = shl i32 %1, 16
%4 = zext i32 %3 to i64
%5 = getelementptr inbounds i32, ptr %0, i64 %4
%6 = load i32, ptr %5, align 4
ret i32 %6
}

define i64 @shl_31_sh3add(ptr %0, i32 signext %1) {
; RV64I-LABEL: shl_31_sh3add:
; RV64I:      # %bb.0:

```

```

; RV64I-NEXT:  slli a1, a1, 31
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  srli a1, a1, 29
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ld a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: shl_31_sh3add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  slli a1, a1, 31
; RV64ZBA-NEXT:  zext.w a1, a1
; RV64ZBA-NEXT:  sh3add a0, a1, a0
; RV64ZBA-NEXT:  ld a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = shl i32 %1, 31
%4 = zext i32 %3
to i64
%5 = getelementptr inbounds i64, ptr %0, i64 %4
%6 = load i64, ptr %5, align 8
ret i64 %6
}

```

```

define i64 @pack_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: pack_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: pack_i64:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  slli a1, a1, 32
; RV64ZBA-NEXT:  add.uw a0, a0, a1
; RV64ZBA-NEXT:  ret
%shl = and i64 %a, 4294967295
%shl1 = shl i64 %b, 32
%or = or i64 %shl1, %shl
ret i64 %or
}

```

```

define i64 @pack_i64_2(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: pack_i64_2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  slli a1, a1, 32

```

```

; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: pack_i64_2:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  slli a1, a1, 32
; RV64ZBA-NEXT:  add.uw a0, a0,
a1
; RV64ZBA-NEXT:  ret
%zexta = zext i32 %a to i64
%zextb = zext i32 %b to i64
%shl1 = shl i64 %zextb, 32
%or = or i64 %shl1, %zexta
ret i64 %or
}

define i64 @pack_i64_3(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: pack_i64_3:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi a0, a0, 1
; RV64I-NEXT:  addi a1, a1, 1
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: pack_i64_3:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  addi a0, a0, 1
; RV64ZBA-NEXT:  addi a1, a1, 1
; RV64ZBA-NEXT:  slli a1, a1, 32
; RV64ZBA-NEXT:  add.uw a0, a0, a1
; RV64ZBA-NEXT:  ret
%adda = add i32 %a, 1
%addb = add i32 %b, 1
%zexta = zext i32 %adda to i64
%zextb = zext i32 %addb to i64
%shl1 = shl i64 %zextb, 32
%or = or i64 %shl1, %zexta
ret i64 %or
}

```

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
Version 2.0, January 2004

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition,

"submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the

same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

#### ---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

=====  
Software from third parties included in the LLVM Project:  
=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or `LICENSE` file at the top containing the specific license and restrictions which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every file.



=====

Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):

=====

The software contained in this directory tree is dual licensed under both the University of Illinois "BSD-Like" license and the MIT license. As a user of this code you may choose to use it under either license. As a contributor, you agree to allow your code to be used under both.

Full text of the relevant licenses is included below.

=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2017-2019 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:  
Threading Runtimes Team  
Intel Corporation  
<http://www.intel.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- \* Neither the names of Intel Corporation Threading Runtimes Team nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====

Copyright (c) 2017-2019 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

IMath is Copyright 2002-2009 Michael J. Fromberger  
You may use it subject to the following Licensing Terms:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -mattr=+zbb -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64ZBB
```

```
declare i32 @llvm.riscv.orc.b.i32(i32)
```

```
define signext i32 @orc32(i32 signext %a) nounwind {
; RV64ZBB-LABEL: orc32:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  orc.b a0, a0
; RV64ZBB-NEXT:  sext.w a0, a0
; RV64ZBB-NEXT:  ret
  %tmp = call i32 @llvm.riscv.orc.b.i32(i32 %a)
  ret i32 %tmp
}
```

```
define zeroext i32 @orc32_zext(i32 zeroext %a) nounwind {
; RV64ZBB-LABEL: orc32_zext:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  orc.b a0, a0
; RV64ZBB-NEXT:  ret
  %tmp = call i32 @llvm.riscv.orc.b.i32(i32 %a)
  ret i32 %tmp
}
```

; Second and+or is redundant with the first, make sure we remove them.

```
define signext i32 @orc32_knownbits(i32 signext %a) nounwind {
; RV64ZBB-LABEL: orc32_knownbits:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  lui a1, 1044480
; RV64ZBB-NEXT:
  and a0, a0, a1
; RV64ZBB-NEXT:  lui a1, 2048
; RV64ZBB-NEXT:  addi a1, a1, 1
; RV64ZBB-NEXT:  or a0, a0, a1
; RV64ZBB-NEXT:  orc.b a0, a0
; RV64ZBB-NEXT:  sext.w a0, a0
; RV64ZBB-NEXT:  ret
  %tmp = and i32 %a, 4278190080 ; 0xFF000000
  %tmp2 = or i32 %tmp, 8388609 ; 0x800001
  %tmp3 = call i32 @llvm.riscv.orc.b.i32(i32 %tmp2)
  %tmp4 = and i32 %tmp3, 4278190080 ; 0xFF000000
  %tmp5 = or i32 %tmp4, 16711935 ; 0xFF00FF
  ret i32 %tmp5
}
```

```
declare i64 @llvm.riscv.orc.b.i64(i64)
```

```

define i64 @orcb64(i64 %a) nounwind {
; RV64ZBB-LABEL: orcb64:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  orc.b a0, a0
; RV64ZBB-NEXT:  ret
%tmp = call i64 @llvm.riscv.orc.b.i64(i64 %a)
ret i64 %tmp
}

```

; Second and+or is redundant with the first, make sure we remove them.

```

define i64 @orcb64_knownbits(i64 %a) nounwind {
; RV64ZBB-LABEL: orcb64_knownbits:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  lui a1, 65535
; RV64ZBB-NEXT:  slli a1, a1, 12
; RV64ZBB-NEXT:  and a0, a0, a1
;
; RV64ZBB-NEXT:  lui a1, 256
; RV64ZBB-NEXT:  addiw a1, a1, 8
; RV64ZBB-NEXT:  slli a2, a1, 42
; RV64ZBB-NEXT:  add a1, a1, a2
; RV64ZBB-NEXT:  or a0, a0, a1
; RV64ZBB-NEXT:  orc.b a0, a0
; RV64ZBB-NEXT:  ret
%tmp = and i64 %a, 1099494850560 ; 0x000000ffff000000
%tmp2 = or i64 %tmp, 4611721202800525320 ; 0x4000200000100008
%tmp3 = call i64 @llvm.riscv.orc.b.i64(i64 %tmp2)
%tmp4 = and i64 %tmp3, 1099494850560 ; 0x000000ffff000000
%tmp5 = or i64 %tmp4, 18374966855153418495 ; 0xff00ff0000ff00ff
ret i64 %tmp5
}

```

; NOTE: Assertions have been autogenerated by utils/update\_llc\_test\_checks.py

; RUN: llc -mtriple=riscv64 -mattr=+zbc -verify-machineinstrs < %s \

; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64ZBC

```

declare i64 @llvm.riscv.clmulr.i64(i64 %a, i64 %b)

```

```

define i64 @clmul64r(i64 %a, i64 %b) nounwind {
; RV64ZBC-LABEL: clmul64r:
; RV64ZBC:      # %bb.0:
; RV64ZBC-NEXT:  clmulr a0, a0, a1
; RV64ZBC-NEXT:  ret
%tmp = call i64 @llvm.riscv.clmulr.i64(i64 %a, i64 %b)
ret i64 %tmp
}

```

```

declare i32 @llvm.riscv.clmulr.i32(i32 %a, i32 %b)

```

```

define signext i32 @clmul32r(i32 signext %a, i32 signext %b) nounwind {
; RV64ZBC-LABEL: clmul32r:
; RV64ZBC:      # %bb.0:
; RV64ZBC-NEXT:  slli a1, a1, 32
; RV64ZBC-NEXT:  slli a0, a0, 32
; RV64ZBC-NEXT:  clmulr a0, a0, a1
; RV64ZBC-NEXT:  srai a0, a0, 32
; RV64ZBC-NEXT:  ret
%tmp = call i32 @llvm.riscv.clmulr.i32(i32 %a, i32 %b)
ret i32 %tmp
}

; FIXME: We could avoid the slli instructions by using clmul+srli+sext.w
; since
; the inputs are zero extended.
define signext i32 @clmul32r_zext(i32 zeroext %a, i32 zeroext %b) nounwind {
; RV64ZBC-LABEL: clmul32r_zext:
; RV64ZBC:      # %bb.0:
; RV64ZBC-NEXT:  slli a1, a1, 32
; RV64ZBC-NEXT:  slli a0, a0, 32
; RV64ZBC-NEXT:  clmulr a0, a0, a1
; RV64ZBC-NEXT:  srai a0, a0, 32
; RV64ZBC-NEXT:  ret
%tmp = call i32 @llvm.riscv.clmulr.i32(i32 %a, i32 %b)
ret i32 %tmp
}

; RUN: opt %s -passes=inline -S | FileCheck %s

define internal void @innerSmall() "min-legal-vector-width"="128" {
ret void
}

define internal void @innerLarge() "min-legal-vector-width"="512" {
ret void
}

define internal void @innerNoAttribute() {
ret void
}

; We should not add an attribute during inlining. No attribute means unknown.
; Inlining doesn't change the fact that we don't know anything about this
; function.
define void @outerNoAttribute() {
call void @innerLarge()
ret void
}

```

```

define void @outerConflictingAttributeSmall() "min-legal-vector-width"="128" {
  call void @innerLarge()
  ret void
}

define void @outerConflictingAttributeLarge() "min-legal-vector-width"="512" {
  call void @innerSmall()
  ret void
}

; We should remove the attribute after inlining since the callee's
; vector width requirements are unknown.
define void @outerAttribute() "min-legal-vector-width"="128" {
  call void @innerNoAttribute()
  ret void
}

; CHECK: define void @outerNoAttribute()
; {
; CHECK: define void @outerConflictingAttributeSmall() #0
; CHECK: define void @outerConflictingAttributeLarge() #0
; CHECK: define void @outerAttribute() {
; CHECK: attributes #0 = { "min-legal-vector-width"="512" }
; RUN: opt -mtriple=aarch64-linux-gnu -mattr=+sve -passes=sanitize-masked-mem-intrin -S < %s | FileCheck %s

; Testing that masked scatters operating on scalable vectors that are
; packed in SVE registers are not scalarized.

; CHECK-LABEL: @masked_scatter_nxv4i32(
; CHECK: call void @llvm.masked.scatter.nxv4i32
define void @masked_scatter_nxv4i32(<vscale x 4 x i32> %data, <vscale x 4 x ptr> %ptrs, <vscale x 4 x i1>
%mask) {
  call void @llvm.masked.scatter.nxv4i32(<vscale x 4 x i32> %data, <vscale x 4 x ptr> %ptrs, i32 0, <vscale x 4 x
i1> %mask)
  ret void
}

; Testing that masked scatters operating on scalable vectors of FP
; data that is packed in SVE registers are not scalarized.

; CHECK-LABEL: @masked_scatter_nxv2f64(
; CHECK: call void @llvm.masked.scatter.nxv2f64
define void @masked_scatter_nxv2f64(<vscale x 2 x double> %data, <vscale x 2 x ptr> %ptrs, <vscale x 2 x i1>
%mask) {
  call void @llvm.masked.scatter.nxv2f64(<vscale x 2 x double> %data, <vscale x 2 x ptr> %ptrs,
i32 0, <vscale x 2 x i1> %mask)
  ret void
}

```

; Testing that masked scatters operating on scalable vectors of FP  
; data that is unpacked in SVE registers are not scalarized.

```
; CHECK-LABEL: @masked_scatter_nxv2f16(  
; CHECK: call void @llvm.masked.scatter.nxv2f16  
define void @masked_scatter_nxv2f16(<vscale x 2 x half> %data, <vscale x 2 x ptr> %ptrs, <vscale x 2 x i1>  
%masks) {  
    call void @llvm.masked.scatter.nxv2f16(<vscale x 2 x half> %data, <vscale x 2 x ptr> %ptrs, i32 0, <vscale x 2 x  
i1> %masks)  
    ret void  
}
```

; Testing that masked scatters operating on 64-bit fixed vectors are  
; scalarized because NEON doesn't have support for masked scatter  
; instructions.

```
; CHECK-LABEL: @masked_scatter_v2f32(  
; CHECK-NOT: @llvm.masked.scatter.v2f32(  
define void @masked_scatter_v2f32(<2 x float> %data, <2 x ptr> %ptrs, <2 x i1> %masks) {  
    call void @llvm.masked.scatter.v2f32(<2 x float> %data, <2 x ptr> %ptrs, i32 0, <2 x i1> %masks)  
    ret void  
}
```

; Testing that masked scatters operating on 128-bit  
fixed vectors are  
; scalarized because NEON doesn't have support for masked scatter  
; instructions and because we are not targeting fixed width SVE.

```
; CHECK-LABEL: @masked_scatter_v4i32(  
; CHECK-NOT: @llvm.masked.scatter.v4i32(  
define void @masked_scatter_v4i32(<4 x i32> %data, <4 x ptr> %ptrs, <4 x i1> %masks) {  
    call void @llvm.masked.scatter.v4i32(<4 x i32> %data, <4 x ptr> %ptrs, i32 0, <4 x i1> %masks)  
    ret void  
}
```

```
declare void @llvm.masked.scatter.nxv4i32(<vscale x 4 x i32> %data, <vscale x 4 x ptr> %ptrs, i32 %align,  
<vscale x 4 x i1> %masks)  
declare void @llvm.masked.scatter.nxv2f64(<vscale x 2 x double> %data, <vscale x 2 x ptr> %ptrs, i32 %align,  
<vscale x 2 x i1> %masks)  
declare void @llvm.masked.scatter.nxv2f16(<vscale x 2 x half> %data, <vscale x 2 x ptr> %ptrs, i32 %align,  
<vscale x 2 x i1> %masks)  
declare void @llvm.masked.scatter.v2f32(<2 x float> %data, <2 x ptr> %ptrs, i32 %align, <2 x i1> %masks)  
declare void @llvm.masked.scatter.v4i32(<4 x i32> %data, <4 x  
ptr> %ptrs, i32 %align, <4 x i1> %masks)  
//===-- ObjectContainerBSDArchive.h -----*- C++ -*-====//  
//  
// Part of the LLVM Project, under the Apache License v2.0 with LLVM Exceptions.
```

```

// See https://llvm.org/LICENSE.txt for license information.
// SPDX-License-Identifier: Apache-2.0 WITH LLVM-exception
//
//===-----
//
#endif
LLDB_SOURCE_PLUGINS_OBJECTCONTAINER_BSD_ARCHIVE_OBJECTCONTAINERBSDARCHIVE_H
#define
LLDB_SOURCE_PLUGINS_OBJECTCONTAINER_BSD_ARCHIVE_OBJECTCONTAINERBSDARCHIVE_H

#include "lldb/Core/UniqueCStringMap.h"
#include "lldb/Symbol/ObjectContainer.h"
#include "lldb/Utility/ArchSpec.h"
#include "lldb/Utility/ConstString.h"
#include "lldb/Utility/FileSpec.h"

#include "llvm/Object/Archive.h"
#include "llvm/Support/Chrono.h"
#include "llvm/Support/Path.h"

#include <map>
#include <memory>
#include <mutex>

enum class ArchiveType { Invalid, Archive, ThinArchive };

class ObjectContainerBSDArchive : public lldb_private::ObjectContainer
{
public:
    ObjectContainerBSDArchive(const lldb::ModuleSP &module_sp,
                             lldb::DataBufferSP &data_sp,
                             lldb::offset_t data_offset,
                             const lldb_private::FileSpec *file,
                             lldb::offset_t offset, lldb::offset_t length,
                             ArchiveType archive_type);

    ~ObjectContainerBSDArchive() override;

    // Static Functions
    static void Initialize();

    static void Terminate();

    static llvm::StringRef GetPluginNameStatic() { return "bsd-archive"; }

    static llvm::StringRef GetPluginDescriptionStatic() {
        return "BSD Archive object container reader.";
    }
}

```



```

static lldb_private::ObjectContainer *
CreateInstance(const lldb::ModuleSP &module_sp, lldb::DataBufferSP &data_sp,
               lldb::offset_t data_offset, const lldb_private::FileSpec *file,
               lldb::offset_t offset, lldb::offset_t length);

static size_t GetModuleSpecifications(const
lldb_private::FileSpec &file,
                                     lldb::DataBufferSP &data_sp,
                                     lldb::offset_t data_offset,
                                     lldb::offset_t file_offset,
                                     lldb::offset_t length,
                                     lldb_private::ModuleSpecList &specs);

static ArchiveType MagicBytesMatch(const lldb_private::DataExtractor &data);

// Member Functions
bool ParseHeader() override;

size_t GetNumObjects() const override {
    if (m_archive_sp)
        return m_archive_sp->GetNumObjects();
    return 0;
}

lldb::ObjectFileSP GetObjectFile(const lldb_private::FileSpec *file) override;

// PluginInterface protocol
llvm::StringRef GetPluginName() override { return GetPluginNameStatic(); }

protected:
struct Object {
    Object();

    void Clear();

    lldb::offset_t ExtractFromThin(const lldb_private::DataExtractor &data,
                                   lldb::offset_t offset,
                                   llvm::StringRef stringTable);

    lldb::offset_t Extract(const lldb_private::DataExtractor &data,
                           lldb::offset_t offset);
    /// Object name in the archive.
    lldb_private::ConstString ar_name;

    /// Object modification time in the archive.
    uint32_t modification_time = 0;

```

```

/// Object size in bytes in the archive.
uint32_t size = 0;

/// File offset in bytes from the beginning of the file of the object data.
lldb::offset_t file_offset = 0;

/// Length of the object data.
lldb::offset_t file_size = 0;

void Dump() const;
};

class Archive {
public:
    typedef std::shared_ptr<Archive> shared_ptr;
    typedef std::multimap<lldb_private::FileSpec, shared_ptr> Map;

    Archive(const lldb_private::ArchSpec &arch,
            const llvm::sys::TimePoint<> &mod_time, lldb::offset_t file_offset,
            lldb_private::DataExtractor &data, ArchiveType archive_type);

    ~Archive();

    static Map &GetArchiveCache();

    static std::recursive_mutex &GetArchiveCacheMutex();

    static Archive::shared_ptr FindCachedArchive(
        const lldb_private::FileSpec &file, const lldb_private::ArchSpec &arch,
        const llvm::sys::TimePoint<> &mod_time, lldb::offset_t file_offset);

    static Archive::shared_ptr ParseAndCacheArchiveForFile(
        const lldb_private::FileSpec &file, const lldb_private::ArchSpec &arch,
        const llvm::sys::TimePoint<> &mod_time, lldb::offset_t file_offset,
        lldb_private::DataExtractor &data, ArchiveType archive_type);

    size_t GetNumObjects() const { return m_objects.size(); }

    const Object *GetObjectAtIndex(size_t idx) {
        if (idx < m_objects.size())
            return &m_objects[idx];
        return nullptr;
    }

    size_t ParseObjects();

    Object *FindObject(lldb_private::ConstString object_name,

```

```

        const llvm::sys::TimePoint<> &object_mod_time);

lldb::offset_t
GetFileOffset() const { return m_file_offset; }

const llvm::sys::TimePoint<> &GetModificationTime() {
    return m_modification_time;
}

const lldb_private::ArchSpec &GetArchitecture() const { return m_arch; }

void SetArchitecture(const lldb_private::ArchSpec &arch) { m_arch = arch; }

bool HasNoExternalReferences() const;

lldb_private::DataExtractor &GetData() { return m_data; }

ArchiveType GetArchiveType() { return m_archive_type; }

protected:
    typedef lldb_private::UniqueCStringMap<uint32_t> ObjectNameToIndexMap;
    // Member Variables
    lldb_private::ArchSpec m_arch;
    llvm::sys::TimePoint<> m_modification_time;
    lldb::offset_t m_file_offset;
    std::vector<Object> m_objects;
    ObjectNameToIndexMap m_object_name_to_index_map;
    lldb_private::DataExtractor m_data; ///< The data for this object container
                                         ///

```

! 2.19.6.2 copyprivate Clause

! If the list item is a polymorphic variable with the allocatable attribute,

! the behavior is unspecified.

subroutine firstprivate()

class(\*), allocatable, save :: x

!PORTABILITY: If a polymorphic variable with allocatable attribute 'x' is in FIRSTPRIVATE clause, the behavior is unspecified

!\$omp parallel firstprivate(x)

call sub()

!\$omp end parallel

end

subroutine lastprivate()

class(\*), allocatable, save :: x

!PORTABILITY: If a polymorphic variable with allocatable attribute 'x' is in LASTPRIVATE clause, the behavior is unspecified

!\$omp do lastprivate(x)

do i = 1, 10

call sub()

enddo

!\$omp end do

end

subroutine copyin()

class(\*), allocatable, save :: x

!\$omp threadprivate(x)

!PORTABILITY: If a polymorphic variable with allocatable attribute 'x' is in COPYIN clause, the behavior is unspecified

!\$omp parallel copyin(x)

call sub()

!\$omp end parallel

end

subroutine copyprivate()

class(\*), allocatable, save :: x

!\$omp threadprivate(x)

!\$omp single

call sub()

!PORTABILITY: If a polymorphic variable with allocatable attribute 'x' is in COPYPRIVATE clause, the behavior is unspecified

!\$omp end single copyprivate(x)

end

=====  
The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:  
=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition,

"submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct

or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.



## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

### ---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions  
of this Software are embedded into an Object form of such source code, you  
may redistribute such embedded portions in such Object form without complying  
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled  
forms of this Software with  
software that is licensed under the GPLv2 ("Combined Software") and if a  
court of competent jurisdiction determines that the patent provision (Section  
3), the indemnity provision (Section 9) or other Section of the License  
conflicts with the conditions of the GPLv2, you may retroactively and  
prospectively choose to deem waived or otherwise exclude such Section(s) of  
the License, but only in their entirety and only with respect to the Combined  
Software.

=====  
Software from third parties included in the LLVM Project:  
=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or `LICENSE` file at the top containing the specific license and restrictions which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every file.

=====

Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):

=====

The libunwind library is dual licensed under both the University of Illinois "BSD-Like" license and the MIT license. As a user of this code you may choose to use it under either license. As a contributor, you agree to allow your code to be used under both.

Full text of the relevant licenses is included below.

=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2009-2019 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

<http://llvm.org>

Permission

is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

\* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.

\* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====

Copyright (c) 2009-2014 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the

Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```
static_library("BSD-Archive") {  
  output_name = "lldbPluginObjectContainerBSDArchive"  
  configs += [ "//llvm/utils/gn/build:lldb_code" ]  
  deps = [  
    "//lldb/source/Core",  
    "//lldb/source/Host",  
    "//lldb/source/Symbol",
```

```

    "//llvm/lib/Support",
]
sources = [ "ObjectContainerBSDArchive.cpp" ]
}
; RUN: opt -mtriple=aarch64-linux-gnu -mattr=+sve -passes=sccp,masked-gather-intrin -S < %s | FileCheck %s

; Testing that masked gathers operating on scalable vectors that are
; packed in SVE registers are not scalarized.

; CHECK-LABEL: @masked_gather_nxv4i32(
; CHECK: call <vscale x 4 x i32> @llvm.masked.gather.nxv4i32
define <vscale x 4 x i32> @masked_gather_nxv4i32(<vscale x 4 x ptr> %ld, <vscale x 4 x i1> %masks, <vscale x 4
x i32> %passthru) {
    %res = call <vscale x 4 x i32> @llvm.masked.gather.nxv4i32(<vscale x 4 x ptr> %ld, i32 0, <vscale x 4 x i1>
    %masks, <vscale x 4 x i32> %passthru)
    ret <vscale x 4 x i32> %res
}

; Testing that masked gathers operating on scalable vectors of FP data
; that is packed in SVE registers are not scalarized.

; CHECK-LABEL: @masked_gather_nxv2f64(
; CHECK: call <vscale x 2 x double> @llvm.masked.gather.nxv2f64
define <vscale x 2 x double> @masked_gather_nxv2f64(<vscale x 2 x ptr> %ld, <vscale x 2 x i1> %masks, <vscale
x 2 x double> %passthru)
{
    %res = call <vscale x 2 x double> @llvm.masked.gather.nxv2f64(<vscale x 2 x ptr> %ld, i32 0, <vscale x 2 x i1>
    %masks, <vscale x 2 x double> %passthru)
    ret <vscale x 2 x double> %res
}

; Testing that masked gathers operating on scalable vectors of FP data
; that is unpacked in SVE registers are not scalarized.

; CHECK-LABEL: @masked_gather_nxv2f16(
; CHECK: call <vscale x 2 x half> @llvm.masked.gather.nxv2f16
define <vscale x 2 x half> @masked_gather_nxv2f16(<vscale x 2 x ptr> %ld, <vscale x 2 x i1> %masks, <vscale x
2 x half> %passthru) {
    %res = call <vscale x 2 x half> @llvm.masked.gather.nxv2f16(<vscale x 2 x ptr> %ld, i32 0, <vscale x 2 x i1>
    %masks, <vscale x 2 x half> %passthru)
    ret <vscale x 2 x half> %res
}

; Testing that masked gathers operating on 64-bit fixed vectors are
; scalarized because NEON doesn't have support for masked gather
; instructions.

; CHECK-LABEL: @masked_gather_v2f32(

```

```
; CHECK-NOT: @llvm.masked.gather.v2f32(
define <2 x float> @masked_gather_v2f32(<2
x ptr> %ld, <2 x i1> %masks, <2 x float> %passthru) {
%res = call <2 x float> @llvm.masked.gather.v2f32(<2 x ptr> %ld, i32 0, <2 x i1> %masks, <2 x float> %passthru)
ret <2 x float> %res
}
```

; Testing that masked gathers operating on 128-bit fixed vectors are  
; scalarized because NEON doesn't have support for masked gather  
; instructions and because we are not targeting fixed width SVE.

```
; CHECK-LABEL: @masked_gather_v4i32(
; CHECK-NOT: @llvm.masked.gather.v4i32(
define <4 x i32> @masked_gather_v4i32(<4 x ptr> %ld, <4 x i1> %masks, <4 x i32> %passthru) {
%res = call <4 x i32> @llvm.masked.gather.v4i32(<4 x ptr> %ld, i32 0, <4 x i1> %masks, <4 x i32> %passthru)
ret <4 x i32> %res
}
```

```
declare <vscale x 4 x i32> @llvm.masked.gather.nxv4i32(<vscale x 4 x ptr> %ptrs, i32 %align, <vscale x 4 x i1>
%masks, <vscale x 4 x i32> %passthru)
declare <vscale x 2 x double> @llvm.masked.gather.nxv2f64(<vscale x 2 x ptr> %ptrs, i32 %align, <vscale x 2 x
i1> %masks, <vscale x 2 x double>
%passthru)
declare <vscale x 2 x half> @llvm.masked.gather.nxv2f16(<vscale x 2 x ptr> %ptrs, i32 %align, <vscale x 2 x i1>
%masks, <vscale x 2 x half> %passthru)
declare <2 x float> @llvm.masked.gather.v2f32(<2 x ptr> %ptrs, i32 %align, <2 x i1> %masks, <2 x float>
%passthru)
declare <4 x i32> @llvm.masked.gather.v4i32(<4 x ptr> %ptrs, i32 %align, <4 x i1> %masks, <4 x i32> %passthru)
# RUN: llc -O0 -mtriple=m68k -start-after=prologuepilg -verify-machineinstrs %s -o - | FileCheck %s
```

name: is-pcrel-register-operand-legal

body: |

bb.0.entry:

```
; CHECK: move.l (0,%pc,%a0), (%a1)
```

```
; CHECK: move.l (%a0), (0,%pc,%a1)
```

```
MOV32jk $a1, 0, $a0, implicit-def $ccr
```

```
MOV32kj 0, $a1, $a0, implicit-def $ccr
```

Ptyprocess is under the ISC license, as code derived from Pexpect.

<http://opensource.org/licenses/ISC>

Copyright (c) 2013-2014, Pexpect development team

Copyright (c) 2012, Noah Spurrier <noah@noah.org>

PERMISSION TO USE, COPY, MODIFY, AND/OR DISTRIBUTE THIS SOFTWARE FOR ANY PURPOSE  
WITH OR WITHOUT FEE IS HEREBY GRANTED, PROVIDED THAT THE ABOVE COPYRIGHT NOTICE  
AND THIS PERMISSION NOTICE APPEAR IN ALL COPIES. THE SOFTWARE IS PROVIDED  
"AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE

INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

UNRECOGNIZED LICENSE; MD5 sum: 2cc784086c8a087e4008cc9f558624bd

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached

to the work

(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition,

"submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.



5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason

of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

### ---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

=====

Software from third parties included in the LLVM Project:

=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or `LICENSE` file at the top containing the specific license and restrictions which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every file.

# People who have agreed to one of the CLAs and can contribute patches.

# The AUTHORS file lists the copyright holders; this file  
# lists people. For example, Google employees are listed here  
# but not in AUTHORS, because Google holds the copyright.  
#

# Names should be added to this file only after verifying that  
# the individual or the individual's organization has agreed to  
# the appropriate Contributor License Agreement, found here:

#  
# <https://developers.google.com/open-source/cla/individual>  
# <https://developers.google.com/open-source/cla/corporate>  
#

# The agreement for individuals can be filled out on the web.  
#

# When adding J Random Contributor's name to this file,  
# either J's name or J's organization's name should be  
# added to the AUTHORS file, depending on whether the  
# individual or corporate CLA was used.

#  
# Names should be added to this file as:

# Name <email address>

#

# Please keep the list sorted.

Abhina Sreeskantharajan <abhina.sreeskantharajan@ibm.com>

Albert Pretorius

<pretoalb@gmail.com>

Alex Steele <steelal123@gmail.com>

Andriy Berestovskyy <berestovskyy@gmail.com>

Arne Beer <arne@twobeer.de>

Billy Robert O'Neal III <billy.oneal@gmail.com> <bion@microsoft.com>

Chris Kennelly <ckennelly@google.com> <ckennelly@ckennelly.com>

Christian Wassermann <christian\_wassermann@web.de>

Christopher Seymour <chris.j.seymour@hotmail.com>

Colin Braley <braley.colin@gmail.com>

Cyrille Faucheux <cyrille.faucheux@gmail.com>

Daniel Harvey <danielharvey458@gmail.com>

David Coeurjolly <david.coeurjolly@liris.cnrs.fr>  
Deniz Evrenci <denizevrenci@gmail.com>  
Dominic Hamon <dma@stripysock.com> <dominic@google.com>  
Dominik Czarnota <dominik.b.czarnota@gmail.com>  
Dominik Korman <kormandominik@gmail.com>  
Donald Aingworth <donalds\_junk\_mail@yahoo.com>  
Eric Backus <eric\_backus@alum.mit.edu>  
Eric Fiselier <eric@efcs.ca>  
Eugene Zhuk <eugene.zhuk@gmail.com>  
Evgeny Safronov <division494@gmail.com>  
Fanbo Meng <fanbo.meng@ibm.com>  
Federico Ficarelli <federico.ficarelli@gmail.com>  
Felix  
Homann <linuxaudio@showlabor.de>  
Geoffrey Martin-Noble <gcmn@google.com> <gmngeoffrey@gmail.com>  
Gerg Szitr <szitar.gergo@gmail.com>  
Hannes Hauswedell <h2@fsfe.org>  
Ismael Jimenez Martinez <ismael.jimenez.martinez@gmail.com>  
Jern-Kuan Leong <jernkuan@gmail.com>  
JianXiong Zhou <zhoujianxiong2@gmail.com>  
Joao Paulo Magalhaes <joaoppmagalhaes@gmail.com>  
John Millikin <jmillikin@stripe.com>  
Jordan Williams <jwillikers@protonmail.com>  
Jussi Knuuttila <jussi.knuuttila@gmail.com>  
Kai Wolf <kai.wolf@gmail.com>  
Kaito Udagawa <umireon@gmail.com>  
Kishan Kumar <kumar.kishan@outlook.com>  
Lei Xu <eddyxu@gmail.com>  
Matt Clarkson <mattyclarkson@gmail.com>  
Maxim Vafin <maxvafin@gmail.com>  
Nick Hutchinson <nshutchinson@gmail.com>  
Norman Heino <norman.heino@gmail.com>  
Oleksandr Sochka <sasha.sochka@gmail.com>  
Ori Livneh <ori.livneh@gmail.com>  
Pascal Leroy <phl@google.com>  
Paul Redmond <paul.redmond@gmail.com>  
Pierre Phaneuf <pphaneuf@google.com>  
Radoslav Yovchev <radoslav.tm@gmail.com>  
Raul Marin  
<rmrodriguez@cartodb.com>  
Ray Glover <ray.glover@uk.ibm.com>  
Robert Guo <robert.guo@mongodb.com>  
Roman Lebedev <lebedev.ri@gmail.com>  
Sayan Bhattacharjee <aero.sayan@gmail.com>  
Shuo Chen <chenshuo@chenshuo.com>  
Steven Wan <wan.yu@ibm.com>  
Tobias Schmidt <tobias.schmidt@in.tum.de>  
Tobias Ulvgrd <tobias.ulvgard@dirac.se>

Tom Madams <tom.ej.madams@gmail.com> <tmadams@google.com>  
Yixuan Qiu <yixuanq@gmail.com>  
Yusuke Suzuki <utatane.tea@gmail.com>  
Zbigniew Skowron <zbychs@gmail.com>  
Min-Yih Hsu <yihshyng223@gmail.com>

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached

to the work

(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition,

"submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative

Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason



of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

### ---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

=====

Software from third parties included in the LLVM Project:

=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or `LICENSE` file at the top containing the specific license and restrictions which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every file.

=====

Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):

=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2007-2018 University of Illinois at Urbana-Champaign.  
All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

<http://llvm.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

\* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.

\* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR

COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

; NOTE: Assertions have been autogenerated by utils/update\_llc\_test\_checks.py

; RUN: llc < %s -mtriple=x86\_64-linux-android -mattr=+mmx -enable-legalize-types-checking | FileCheck %s

;

; D31946

; Check that we dont end up with the ""LLVM ERROR: Cannot select" error.

; Additionally ensure that the output code actually put fp128 values in SSE registers.

declare fp128 @llvm.fabs.f128(fp128)

declare fp128 @llvm.copysign.f128(fp128, fp128)

define fp128 @TestSelect(fp128 %a, fp128 %b) {

; CHECK-LABEL: TestSelect:

; CHECK: # %bb.0:

; CHECK-NEXT: pushq %rbx

; CHECK-NEXT: .cfi\_def\_cfa\_offset 16

; CHECK-NEXT: subq \$32, %rsp

; CHECK-NEXT: .cfi\_def\_cfa\_offset 48

; CHECK-NEXT: .cfi\_offset %rbx, -16

; CHECK-NEXT: movaps %xmm1, {[[-0-9]+]}(%r{[sb]})p) # 16-byte Spill

; CHECK-NEXT: movaps %xmm0, (%rsp) # 16-byte Spill

; CHECK-NEXT: callq \_\_gttf2@PLT

; CHECK-NEXT: movl %eax, %ebx

; CHECK-NEXT: movaps (%rsp), %xmm0 # 16-byte Reload

; CHECK-NEXT: movaps {[[-0-9]+]}(%r{[sb]})p),

%xmm1 # 16-byte Reload

; CHECK-NEXT: callq \_\_subtf3@PLT

; CHECK-NEXT: testl %ebx, %ebx

; CHECK-NEXT: jg .LBB0\_2

; CHECK-NEXT: # %bb.1:

; CHECK-NEXT: xorps %xmm0, %xmm0

; CHECK-NEXT: .LBB0\_2:

; CHECK-NEXT: addq \$32, %rsp

; CHECK-NEXT: .cfi\_def\_cfa\_offset 16

; CHECK-NEXT: popq %rbx

; CHECK-NEXT: .cfi\_def\_cfa\_offset 8

; CHECK-NEXT: retq

%cmp = fcmp ogt fp128 %a, %b

%sub = fsub fp128 %a, %b

%res = select i1 %cmp, fp128 %sub, fp128 0xL00000000000000000000000000000000

ret fp128 %res

}

define fp128 @TestFabs(fp128 %a) {

; CHECK-LABEL: TestFabs:

```
; CHECK:      # %bb.0:
; CHECK-NEXT:  andps {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %xmm0
; CHECK-NEXT:  retq
%res = call fp128 @llvm.fabs.f128(fp128 %a)
ret fp128 %res
}
```

```
define fp128 @TestCopysign(fp128 %a, fp128 %b) {
; CHECK-LABEL: TestCopysign:
; CHECK:      # %bb.0:
; CHECK-NEXT:  andps {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %xmm1
; CHECK-NEXT:  andps {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %xmm0
; CHECK-NEXT:
    orps %xmm1, %xmm0
; CHECK-NEXT:  retq
%res = call fp128 @llvm.copysign.f128(fp128 %a, fp128 %b)
ret fp128 %res
}
```

```
define fp128 @TestFneg(fp128 %a) {
; CHECK-LABEL: TestFneg:
; CHECK:      # %bb.0:
; CHECK-NEXT:  pushq %rax
; CHECK-NEXT:  .cfi_def_cfa_offset 16
; CHECK-NEXT:  movaps %xmm0, %xmm1
; CHECK-NEXT:  callq __multf3@PLT
; CHECK-NEXT:  xorps {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %xmm0
; CHECK-NEXT:  popq %rax
; CHECK-NEXT:  .cfi_def_cfa_offset 8
; CHECK-NEXT:  retq
%mul = fmul fp128 %a, %a
%res = fsub fp128 0xL00000000000000008000000000000000, %mul
ret fp128 %res
}
```

---

## clang-tidy CERT Files

---

All clang-tidy files are licensed under the same terms as the rest of the LLVM project with the following additions:

Any file referencing a CERT Secure Coding guideline:

Please allow this letter to serve as confirmation that open source projects on <http://llvm.org> are permitted to link via hypertext to the CERT(R) secure coding guidelines available at <https://www.securecoding.cert.org>.

The foregoing is permitted by the Terms of Use as follows:

"Linking to the Service

Because we update many of our Web documents regularly, we would prefer that you

link to our Web pages whenever possible rather than reproduce them. It is not necessary to request permission to make referential hypertext links to The Service."

<http://www.sei.cmu.edu/legal/ip/index.cfm>.

Please allow this letter to also confirm that no formal permission is required

to reproduce the title of the content being linked to, nor to reproduce any de Minimis description of such content.

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or

for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License

does not grant permission to use the trade

names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf



of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

### ---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined

Software.

=====

Software from third parties included in the LLVM Project:

=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or `LICENSE` file at the top containing the specific license and restrictions which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every file.

=====

Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):

=====

The libclc library is dual licensed under both the University of Illinois "BSD-Like" license and the MIT license. As a user of this code you may choose to use it under either license. As a contributor, you agree to allow your code to be used under both.

Full text of the relevant licenses is included below.

=====

Copyright (c) 2011-2019 by the contributors listed in CREDITS.TXT

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- \* The names of the contributors may not be used to endorse or promote

products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR

PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====

Copyright (c) 2011-2014 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE

SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

; NOTE: Assertions have been autogenerated by utils/update\_llc\_test\_checks.py

; RUN: llc -mtriple=riscv64 -mattr=+zbbk -verify-machineinstrs < %s \

; RUN: | FileCheck %s -check-prefix=RV64ZBKB

declare i64 @llvm.riscv.brev8.i64(i64)

```
define i64 @brev8(i64 %a) nounwind {
; RV64ZBKB-LABEL: brev8:
; RV64ZBKB:      # %bb.0:
; RV64ZBKB-NEXT:  brev8 a0, a0
; RV64ZBKB-NEXT:  ret
%val = call i64 @llvm.riscv.brev8.i64(i64 %a)
ret i64 %val
}
```

```

; Test that rev8 is recognized as preserving zero extension.
define zeroext i16 @brev8_knownbits(i16 zeroext %a) nounwind {
; RV64ZBKB-LABEL: brev8_knownbits:
; RV64ZBKB:      # %bb.0:
; RV64ZBKB-NEXT:  brev8 a0, a0
; RV64ZBKB-NEXT:  ret
%zext = zext i16 %a to i64
%val = call i64 @llvm.riscv.brev8.i64(i64 %zext)
%trunc = trunc i64 %val to i16
ret i16 %trunc
}

declare i64 @llvm.bswap.i64(i64)

define i64 @rev8_i64(i64 %a) {
; RV64ZBKB-LABEL: rev8_i64:
; RV64ZBKB:      # %bb.0:
; RV64ZBKB-NEXT:  rev8 a0, a0
; RV64ZBKB-NEXT:  ret
%1 = call i64 @llvm.bswap.i64(i64 %a)
ret i64 %1
}

declare i32 @llvm.riscv.brev8.i32(i32)

define signext i32 @brev8_i32(i32 signext %a) nounwind {
; RV64ZBKB-LABEL: brev8_i32:
; RV64ZBKB:      # %bb.0:
; RV64ZBKB-NEXT:  brev8 a0, a0
; RV64ZBKB-NEXT:  sext.w a0, a0
; RV64ZBKB-NEXT:  ret
%val = call i32 @llvm.riscv.brev8.i32(i32 %a)
ret i32 %val
}

; Test that rev8 is recognized as preserving zero extension.
define zeroext i16 @brev8_i32_knownbits(i16 zeroext %a) nounwind {
; RV64ZBKB-LABEL: brev8_i32_knownbits:
; RV64ZBKB:      # %bb.0:
; RV64ZBKB-NEXT:  brev8 a0, a0
; RV64ZBKB-NEXT:  ret
%zext = zext i16 %a to i32
%val = call i32 @llvm.riscv.brev8.i32(i32 %zext)
%trunc = trunc i32 %val to i16
ret i16 %trunc
}

```

```
declare i32 @llvm.bswap.i32(i32)
```

```
define signext i32 @rev8_i32(i32 signext %a) {  
; RV64ZBKB-LABEL: rev8_i32:  
; RV64ZBKB:      # %bb.0:  
; RV64ZBKB-NEXT: rev8 a0, a0  
; RV64ZBKB-NEXT: srai a0, a0, 32  
; RV64ZBKB-NEXT: ret  
%1 = call i32 @llvm.bswap.i32(i32  
%a)  
ret i32 %1  
}
```

; NOTE: Assertions have been autogenerated by utils/update\_llc\_test\_checks.py

; RUN: llc -mtriple=amdgcn -mcpu=fiji < %s | FileCheck -check-prefixes=VI %s

; FIXME: This one should fold to rcp

```
define half @select_fneg_posk_src_rcp_f16(i32 %c, half %x, half %y) {  
; VI-LABEL: select_fneg_posk_src_rcp_f16:  
; VI:      ; %bb.0:  
; VI-NEXT: s_waitcnt vmcnt(0) expcnt(0) lgkmcnt(0)  
; VI-NEXT: v_rcp_f16_e64 v1, -v1  
; VI-NEXT: v_mov_b32_e32 v2, 0x4000  
; VI-NEXT: v_cmp_eq_u32_e32 vcc, 0, v0  
; VI-NEXT: v_cndmask_b32_e32 v0, v2, v1, vcc  
; VI-NEXT: s_setpc_b64 s[30:31]  
%cmp = icmp eq i32 %c, 0  
%rcp = call half @llvm.amdgcn.rcp.f16(half %x)  
%fneg = fneg half %rcp  
%select = select i1 %cmp, half %fneg, half 2.0  
ret half %select  
}
```

```
declare half @llvm.amdgcn.rcp.f16(half) #0
```

attributes #0 = { nocallback nofree nosync nounwind speculatable willreturn memory(none) }

Apache License 2.0

; NOTE: Assertions have been autogenerated by utils/update\_llc\_test\_checks.py

; RUN: llc -mtriple=riscv64 -verify-machineinstrs < %s \

; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefixes=CHECK,RV64I

; RUN: llc -mtriple=riscv64 -mattr=+zbb -verify-machineinstrs < %s \

; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefixes=CHECK,RV64ZBB-ZBKB,RV64ZBB

; RUN: llc -mtriple=riscv64 -mattr=+zbbk -verify-machineinstrs < %s \

; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefixes=CHECK,RV64ZBB-ZBKB,RV64ZBKB

```
define signext i32 @andn_i32(i32 signext %a, i32 signext %b) nounwind {  
; RV64I-LABEL: andn_i32:  
; RV64I:      # %bb.0:
```

```

; RV64I-NEXT:  not a1, a1
; RV64I-NEXT:  and a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: andn_i32:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  andn a0, a0, a1
; RV64ZBB-ZBKB-NEXT:  ret
%neg = xor i32 %b, -1
%and = and i32 %neg, %a
ret i32 %and
}

define i64 @andn_i64(i64 %a, i64
%b) nounwind {
; RV64I-LABEL: andn_i64:
; RV64I:    # %bb.0:
; RV64I-NEXT:  not a1, a1
; RV64I-NEXT:  and a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: andn_i64:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  andn a0, a0, a1
; RV64ZBB-ZBKB-NEXT:  ret
%neg = xor i64 %b, -1
%and = and i64 %neg, %a
ret i64 %and
}

define signext i32 @orn_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: orn_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT:  not a1, a1
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: orn_i32:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  orn a0, a0, a1
; RV64ZBB-ZBKB-NEXT:  ret
%neg = xor i32 %b, -1
%or = or i32 %neg, %a
ret i32 %or
}

define i64 @orn_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: orn_i64:

```

```

; RV64I:    # %bb.0:
; RV64I-NEXT:  not a1, a1
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: orn_i64:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:
    orn a0, a0, a1
; RV64ZBB-ZBKB-NEXT:  ret
    %neg = xor i64 %b, -1
    %or = or i64 %neg, %a
    ret i64 %or
}

define signext i32 @xnor_i32(i32 signext %a, i32 signext %b) nounwind {
; CHECK-LABEL: xnor_i32:
; CHECK:    # %bb.0:
; CHECK-NEXT:  xor a0, a0, a1
; CHECK-NEXT:  not a0, a0
; CHECK-NEXT:  ret
    %neg = xor i32 %a, -1
    %xor = xor i32 %neg, %b
    ret i32 %xor
}

define i64 @xnor_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: xnor_i64:
; RV64I:    # %bb.0:
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: xnor_i64:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  xnor a0, a0, a1
; RV64ZBB-ZBKB-NEXT:  ret
    %neg = xor i64 %a, -1
    %xor = xor i64 %neg, %b
    ret i64 %xor
}

declare i32 @llvm.fshl.i32(i32, i32, i32)

define signext i32 @rol_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: rol_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT:  andi a2, a1, -1

```

```

; RV64I-NEXT:  sllw a1, a0, a1
; RV64I-NEXT:
    negw a2, a2
; RV64I-NEXT:  srlw a0, a0, a2
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: rol_i32:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  rolw a0, a0, a1
; RV64ZBB-ZBKB-NEXT:  ret
%1 = tail call i32 @llvm.fshl.i32(i32 %a, i32 %a, i32 %b)
ret i32 %1
}

; Similar to rol_i32, but doesn't sign extend the result.
define void @rol_i32_nosext(i32 signext %a, i32 signext %b, ptr %x) nounwind {
; RV64I-LABEL: rol_i32_nosext:
; RV64I:    # %bb.0:
; RV64I-NEXT:  andi a3, a1, -1
; RV64I-NEXT:  sllw a1, a0, a1
; RV64I-NEXT:  negw a3, a3
; RV64I-NEXT:  srlw a0, a0, a3
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  sw a0, 0(a2)
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: rol_i32_nosext:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  rolw a0, a0, a1
; RV64ZBB-ZBKB-NEXT:  sw a0, 0(a2)
; RV64ZBB-ZBKB-NEXT:  ret
%1 = tail call i32 @llvm.fshl.i32(i32 %a, i32 %a, i32 %b)
store i32 %1, ptr %x
ret void
}

define signext
i32 @rol_i32_neg_constant_rhs(i32 signext %a) nounwind {
; RV64I-LABEL: rol_i32_neg_constant_rhs:
; RV64I:    # %bb.0:
; RV64I-NEXT:  andi a1, a0, -1
; RV64I-NEXT:  li a2, -2
; RV64I-NEXT:  sllw a0, a2, a0
; RV64I-NEXT:  negw a1, a1
; RV64I-NEXT:  srlw a1, a2, a1
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret

```



```

;
; RV64ZBB-ZBKB-LABEL: rol_i32_neg_constant_rhs:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  li a1, -2
; RV64ZBB-ZBKB-NEXT:  rolw a0, a1, a0
; RV64ZBB-ZBKB-NEXT:  ret
%1 = tail call i32 @llvm.fshl.i32(i32 -2, i32 -2, i32 %a)
ret i32 %1
}

```

```

declare i64 @llvm.fshl.i64(i64, i64, i64)

```

```

define i64 @rol_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: rol_i64:
; RV64I:    # %bb.0:
; RV64I-NEXT:  sll a2, a0, a1
; RV64I-NEXT:  negw a1, a1
; RV64I-NEXT:  srl a0, a0, a1
; RV64I-NEXT:  or a0, a2, a0
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: rol_i64:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  rol a0, a0, a1
; RV64ZBB-ZBKB-NEXT:
ret
%or = tail call i64 @llvm.fshl.i64(i64 %a, i64 %a, i64 %b)
ret i64 %or
}

```

```

declare i32 @llvm.fshr.i32(i32, i32, i32)

```

```

define signext i32 @ror_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: ror_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT:  andi a2, a1, -1
; RV64I-NEXT:  srlw a1, a0, a1
; RV64I-NEXT:  negw a2, a2
; RV64I-NEXT:  sllw a0, a0, a2
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: ror_i32:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  rorw a0, a0, a1
; RV64ZBB-ZBKB-NEXT:  ret
%1 = tail call i32 @llvm.fshr.i32(i32 %a, i32 %a, i32 %b)
ret i32 %1

```

```
}
```

; Similar to ror\_i32, but doesn't sign extend the result.

```
define void @ror_i32_nosext(i32 signext %a, i32 signext %b, ptr %x) nounwind {
; RV64I-LABEL: ror_i32_nosext:
; RV64I:      # %bb.0:
; RV64I-NEXT: andi a3, a1, -1
; RV64I-NEXT: srlw a1, a0, a1
; RV64I-NEXT: negw a3, a3
; RV64I-NEXT: sllw a0, a0, a3
; RV64I-NEXT: or a0, a1, a0
; RV64I-NEXT:
;               sw a0, 0(a2)
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: ror_i32_nosext:
; RV64ZBB-ZBKB:      # %bb.0:
; RV64ZBB-ZBKB-NEXT: rorw a0, a0, a1
; RV64ZBB-ZBKB-NEXT: sw a0, 0(a2)
; RV64ZBB-ZBKB-NEXT: ret
%1 = tail call i32 @llvm.fshr.i32(i32 %a, i32 %a, i32 %b)
store i32 %1, ptr %x
ret void
}
```

```
define signext i32 @ror_i32_neg_constant_rhs(i32 signext %a) nounwind {
; RV64I-LABEL: ror_i32_neg_constant_rhs:
; RV64I:      # %bb.0:
; RV64I-NEXT: andi a1, a0, -1
; RV64I-NEXT: li a2, -2
; RV64I-NEXT: srlw a0, a2, a0
; RV64I-NEXT: negw a1, a1
; RV64I-NEXT: sllw a1, a2, a1
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: ret
;
; RV64ZBB-ZBKB-LABEL: ror_i32_neg_constant_rhs:
; RV64ZBB-ZBKB:      # %bb.0:
; RV64ZBB-ZBKB-NEXT: li a1, -2
; RV64ZBB-ZBKB-NEXT: rorw a0, a1, a0
; RV64ZBB-ZBKB-NEXT: ret
%1 = tail call i32 @llvm.fshr.i32(i32 -2, i32 -2, i32 %a)
ret i32 %1
}
```

```
declare i64 @llvm.fshr.i64(i64, i64, i64)
```

```
define i64 @ror_i64(i64 %a, i64 %b) nounwind {
```

```

;
RV64I-LABEL: ror_i64:
; RV64I:    # %bb.0:
; RV64I-NEXT:  srl a2, a0, a1
; RV64I-NEXT:  negw a1, a1
; RV64I-NEXT:  sll a0, a0, a1
; RV64I-NEXT:  or a0, a2, a0
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: ror_i64:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  ror a0, a0, a1
; RV64ZBB-ZBKB-NEXT:  ret
%or = tail call i64 @llvm.fshr.i64(i64 %a, i64 %a, i64 %b)
ret i64 %or
}

define signext i32 @rori_i32_fshl(i32 signext %a) nounwind {
; RV64I-LABEL: rori_i32_fshl:
; RV64I:    # %bb.0:
; RV64I-NEXT:  srlw a1, a0, 1
; RV64I-NEXT:  sllw a0, a0, 31
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: rori_i32_fshl:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  roriw a0, a0, 1
; RV64ZBB-ZBKB-NEXT:  ret
%1 = tail call i32 @llvm.fshl.i32(i32 %a, i32 %a, i32 31)
ret i32 %1
}

; Similar to rori_i32_fshl, but doesn't sign extend the result.
define void @rori_i32_fshl_nosext(i32 signext %a, ptr %x) nounwind {
; RV64I-LABEL: rori_i32_fshl_nosext:
;
RV64I:    # %bb.0:
; RV64I-NEXT:  srlw a2, a0, 1
; RV64I-NEXT:  slli a0, a0, 31
; RV64I-NEXT:  or a0, a0, a2
; RV64I-NEXT:  sw a0, 0(a1)
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: rori_i32_fshl_nosext:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  roriw a0, a0, 1
; RV64ZBB-ZBKB-NEXT:  sw a0, 0(a1)

```

```

; RV64ZBB-ZBKB-NEXT:  ret
%1 = tail call i32 @llvm.fshl.i32(i32 %a, i32 %a, i32 31)
store i32 %1, ptr %x
ret void
}

define signext i32 @rori_i32_fshr(i32 signext %a) nounwind {
; RV64I-LABEL: rori_i32_fshr:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slliw a1, a0, 1
; RV64I-NEXT:  srliw a0, a0, 31
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: rori_i32_fshr:
; RV64ZBB-ZBKB:  # %bb.0:
; RV64ZBB-ZBKB-NEXT:  roriw a0, a0, 31
; RV64ZBB-ZBKB-NEXT:  ret
%1 = tail call i32 @llvm.fshr.i32(i32 %a, i32 %a, i32 31)
ret i32 %1
}

```

; Similar to rori\_i32\_fshr, but doesn't sign extend the result.

```

define void @rori_i32_fshr_nosext(i32
signext %a, ptr %x) nounwind {
; RV64I-LABEL: rori_i32_fshr_nosext:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a2, a0, 1
; RV64I-NEXT:  srliw a0, a0, 31
; RV64I-NEXT:  or a0, a0, a2
; RV64I-NEXT:  sw a0, 0(a1)
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: rori_i32_fshr_nosext:
; RV64ZBB-ZBKB:  # %bb.0:
; RV64ZBB-ZBKB-NEXT:  roriw a0, a0, 31
; RV64ZBB-ZBKB-NEXT:  sw a0, 0(a1)
; RV64ZBB-ZBKB-NEXT:  ret
%1 = tail call i32 @llvm.fshr.i32(i32 %a, i32 %a, i32 31)
store i32 %1, ptr %x
ret void
}

```

; This test is similar to the type legalized version of the fshl/fshr tests, but

; instead of having the same input to both shifts it has different inputs. Make

; sure we don't match it as a roriw.

```

define signext i32 @not_rori_i32(i32 signext %x, i32 signext %y) nounwind {
; CHECK-LABEL: not_rori_i32:

```

```

; CHECK:      # %bb.0:
; CHECK-NEXT:  slliw a0, a0, 31
; CHECK-NEXT:  srliw a1, a1, 1
; CHECK-NEXT:  or a0, a0, a1
; CHECK-NEXT:  ret
%a = shl
i32 %x, 31
%b = lshr i32 %y, 1
%c = or i32 %a, %b
ret i32 %c
}

```

; This is similar to the type legalized roriw pattern, but the and mask is more  
; than 32 bits so the lshr doesn't shift zeroes into the lower 32 bits. Make  
; sure we don't match it to roriw.

```

define i64 @roriw_bug(i64 %x) nounwind {
; CHECK-LABEL: roriw_bug:
; CHECK:      # %bb.0:
; CHECK-NEXT:  slli a1, a0, 31
; CHECK-NEXT:  andi a2, a0, -2
; CHECK-NEXT:  srli a0, a0, 1
; CHECK-NEXT:  or a0, a1, a0
; CHECK-NEXT:  sext.w a0, a0
; CHECK-NEXT:  xor a0, a2, a0
; CHECK-NEXT:  ret
%a = shl i64 %x, 31
%b = and i64 %x, 18446744073709551614
%c = lshr i64 %b, 1
%d = or i64 %a, %c
%e = shl i64 %d, 32
%f = ashr i64 %e, 32
%g = xor i64 %b, %f ; to increase the use count on %b to disable SimplifyDandedBits.
ret i64 %g
}

```

```

define i64 @rori_i64_fshl(i64 %a) nounwind {
; RV64I-LABEL: rori_i64_fshl:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srli a1, a0, 1
; RV64I-NEXT:  slli a0, a0, 63
; RV64I-NEXT:
;   or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: rori_i64_fshl:
; RV64ZBB-ZBKB:  # %bb.0:
; RV64ZBB-ZBKB-NEXT:  rori a0, a0, 1
; RV64ZBB-ZBKB-NEXT:  ret

```

```

%1 = tail call i64 @llvm.fshl.i64(i64 %a, i64 %a, i64 63)
ret i64 %1
}

define i64 @rori_i64_fshr(i64 %a) nounwind {
; RV64I-LABEL: rori_i64_fshr:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a1, a0, 1
; RV64I-NEXT:  srli a0, a0, 63
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: rori_i64_fshr:
; RV64ZBB-ZBKB:  # %bb.0:
; RV64ZBB-ZBKB-NEXT:  rori a0, a0, 63
; RV64ZBB-ZBKB-NEXT:  ret
%1 = tail call i64 @llvm.fshr.i64(i64 %a, i64 %a, i64 63)
ret i64 %1
}

define signext i32 @not_shl_one_i32(i32 signext %x) {
; RV64I-LABEL: not_shl_one_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 1
; RV64I-NEXT:  sllw a0, a1, a0
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: not_shl_one_i32:
; RV64ZBB-ZBKB:  # %bb.0:
; RV64ZBB-ZBKB-NEXT:  li a1,
-2
; RV64ZBB-ZBKB-NEXT:  rolw a0, a1, a0
; RV64ZBB-ZBKB-NEXT:  ret
%1 = shl i32 1, %x
%2 = xor i32 %1, -1
ret i32 %2
}

define i64 @not_shl_one_i64(i64 %x) {
; RV64I-LABEL: not_shl_one_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 1
; RV64I-NEXT:  sll a0, a1, a0
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: not_shl_one_i64:

```

```
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  li a1, -2
; RV64ZBB-ZBKB-NEXT:  rol a0, a1, a0
; RV64ZBB-ZBKB-NEXT:  ret
%1 = shl i64 1, %x
%2 = xor i64 %1, -1
ret i64 %2
}
```

```
define i8 @srli_i8(i8 %a) nounwind {
; CHECK-LABEL: srli_i8:
; CHECK:      # %bb.0:
; CHECK-NEXT:  slli a0, a0, 56
; CHECK-NEXT:  srli a0, a0, 62
; CHECK-NEXT:  ret
%1 = lshr i8 %a, 6
ret i8 %1
}
```

; We could use sext.b+srai, but slli+srai offers more opportunities for  
; compressed instructions.

```
define i8 @srai_i8(i8 %a) nounwind {
; RV64I-LABEL: srai_i8:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 24
; RV64I-NEXT:
    sraiw a0, a0, 29
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: srai_i8:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT:  slli a0, a0, 56
; RV64ZBB-NEXT:  srai a0, a0, 61
; RV64ZBB-NEXT:  ret
;
; RV64ZBKB-LABEL: srai_i8:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  slli a0, a0, 24
; RV64ZBKB-NEXT:  sraiw a0, a0, 29
; RV64ZBKB-NEXT:  ret
%1 = ashr i8 %a, 5
ret i8 %1
}
```

; We could use zext.h+srli, but slli+srli offers more opportunities for  
; compressed instructions.

```
define i16 @srli_i16(i16 %a) nounwind {
; CHECK-LABEL: srli_i16:
```

```
; CHECK:    # %bb.0:
; CHECK-NEXT:  slli a0, a0, 48
; CHECK-NEXT:  srli a0, a0, 54
; CHECK-NEXT:  ret
%1 = lshr i16 %a, 6
ret i16 %1
}
```

; We could use sext.h+srai, but slli+srai offers more opportunities for  
; compressed instructions.

```
define i16 @srai_i16(i16 %a) nounwind {
```

```
; RV64I-LABEL: srai_i16:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 16
; RV64I-NEXT:  sraiw a0, a0, 25
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: srai_i16:
;
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT:  slli a0, a0, 48
; RV64ZBB-NEXT:  srai a0, a0, 57
; RV64ZBB-NEXT:  ret
;
; RV64ZBKB-LABEL: srai_i16:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  slli a0, a0, 16
; RV64ZBKB-NEXT:  sraiw a0, a0, 25
; RV64ZBKB-NEXT:  ret
%1 = ashr i16 %a, 9
ret i16 %1
}
```

```
define i1 @andn_seqz_i32(i32 signext %a, i32 signext %b) nounwind {
```

```
; RV64I-LABEL: andn_seqz_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  seqz a0, a0
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: andn_seqz_i32:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  andn a0, a1, a0
; RV64ZBB-ZBKB-NEXT:  seqz a0, a0
; RV64ZBB-ZBKB-NEXT:  ret
%and = and i32 %a, %b
%cmpeq = icmp eq i32 %and, %b
```



```

ret i1 %cmpeq
}

define i1 @andn_seqz_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: andn_seqz_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  seqz a0, a0
; RV64I-NEXT:
    ret
;
; RV64ZBB-ZBKB-LABEL: andn_seqz_i64:
; RV64ZBB-ZBKB:  # %bb.0:
; RV64ZBB-ZBKB-NEXT:  andn a0, a1, a0
; RV64ZBB-ZBKB-NEXT:  seqz a0, a0
; RV64ZBB-ZBKB-NEXT:  ret
    %and = and i64 %a, %b
    %cmpeq = icmp eq i64 %and, %b
    ret i1 %cmpeq
}

define i1 @andn_snez_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: andn_snez_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  snez a0, a0
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: andn_snez_i32:
; RV64ZBB-ZBKB:  # %bb.0:
; RV64ZBB-ZBKB-NEXT:  andn a0, a1, a0
; RV64ZBB-ZBKB-NEXT:  snez a0, a0
; RV64ZBB-ZBKB-NEXT:  ret
    %and = and i32 %a, %b
    %cmpeq = icmp ne i32 %and, %b
    ret i1 %cmpeq
}

define i1 @andn_snez_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: andn_snez_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  snez a0, a0
; RV64I-NEXT:  ret
;

```

```

; RV64ZBB-ZBKB-LABEL: andn_snez_i64:
;
RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  andn a0, a1, a0
; RV64ZBB-ZBKB-NEXT:  snez a0, a0
; RV64ZBB-ZBKB-NEXT:  ret
%and = and i64 %a, %b
%cmpeq = icmp ne i64 %and, %b
ret i1 %cmpeq
}
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck -check-prefix=RV64I %s
; RUN: llc -mtriple=riscv64 -mattr=+m -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck -check-prefix=RV64IM %s

```

```

define i32 @urem(i32 %a, i32 %b) nounwind {
; RV64I-LABEL: urem:
; RV64I:    # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  srli a1, a1, 32
; RV64I-NEXT:  call __umoddi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: urem:
; RV64IM:    # %bb.0:
; RV64IM-NEXT:  remuw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = urem i32 %a, %b
ret i32 %1
}

```

```

define i32 @urem_constant_lhs(i32 %a) nounwind {
;
RV64I-LABEL: urem_constant_lhs:
; RV64I:    # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a1, a0, 32
; RV64I-NEXT:  li a0, 10
; RV64I-NEXT:  call __umoddi3

```

```

; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: urem_constant_lhs:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  li a1, 10
; RV64IM-NEXT:  remuw a0, a1, a0
; RV64IM-NEXT:  ret
%1 = urem i32 10, %a
ret i32 %1
}

```

```

define i32 @srem(i32 %a, i32 %b) nounwind {
; RV64I-LABEL: srem:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  sext.w a0, a0
; RV64I-NEXT:  sext.w a1, a1
; RV64I-NEXT:  call __moddi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: srem:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:
    remw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = srem i32 %a, %b
ret i32 %1
}

```

```

define i32 @srem_pow2(i32 %a) nounwind {
; RV64I-LABEL: srem_pow2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  sraiw a1, a0, 31
; RV64I-NEXT:  srliw a1, a1, 29
; RV64I-NEXT:  add a1, a0, a1
; RV64I-NEXT:  andi a1, a1, -8
; RV64I-NEXT:  subw a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: srem_pow2:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  sraiw a1, a0, 31
; RV64IM-NEXT:  srliw a1, a1, 29

```

```

; RV64IM-NEXT:  add a1, a0, a1
; RV64IM-NEXT:  andi a1, a1, -8
; RV64IM-NEXT:  subw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = srem i32 %a, 8
ret i32 %1
}

```

```

define i32 @srem_pow2_2(i32 %a) nounwind {
; RV64I-LABEL: srem_pow2_2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  sraiw a1, a0, 31
; RV64I-NEXT:  srlw a1, a1, 16
; RV64I-NEXT:  add a1, a0, a1
; RV64I-NEXT:  lui a2, 1048560
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  subw a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: srem_pow2_2:
; RV64IM:
;   # %bb.0:
; RV64IM-NEXT:  sraiw a1, a0, 31
; RV64IM-NEXT:  srlw a1, a1, 16
; RV64IM-NEXT:  add a1, a0, a1
; RV64IM-NEXT:  lui a2, 1048560
; RV64IM-NEXT:  and a1, a1, a2
; RV64IM-NEXT:  subw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = srem i32 %a, 65536
ret i32 %1
}

```

```

define i32 @srem_constant_lhs(i32 %a) nounwind {
; RV64I-LABEL: srem_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  sext.w a1, a0
; RV64I-NEXT:  li a0, -10
; RV64I-NEXT:  call __moddi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: srem_constant_lhs:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  li a1, -10

```

```

; RV64IM-NEXT:  remw a0, a1, a0
; RV64IM-NEXT:  ret
%1 = srem i32 -10, %a
ret i32 %1
}

define i64 @urem64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: urem64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  tail __umoddi3
;
; RV64IM-LABEL:
urem64:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  remu a0, a0, a1
; RV64IM-NEXT:  ret
%1 = urem i64 %a, %b
ret i64 %1
}

define i64 @urem64_constant_lhs(i64 %a) nounwind {
; RV64I-LABEL: urem64_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  mv a1, a0
; RV64I-NEXT:  li a0, 10
; RV64I-NEXT:  tail __umoddi3
;
; RV64IM-LABEL: urem64_constant_lhs:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  li a1, 10
; RV64IM-NEXT:  remu a0, a1, a0
; RV64IM-NEXT:  ret
%1 = urem i64 10, %a
ret i64 %1
}

define i64 @srem64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: srem64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  tail __moddi3
;
; RV64IM-LABEL: srem64:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  rem a0, a0, a1
; RV64IM-NEXT:  ret
%1 = srem i64 %a, %b
ret i64 %1
}

```

```

define i64 @srem64_constant_lhs(i64 %a) nounwind {
; RV64I-LABEL: srem64_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  mv a1, a0
; RV64I-NEXT:  li a0, -10
; RV64I-NEXT:  tail __moddi3
;
; RV64IM-LABEL: srem64_constant_lhs:
;
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  li a1, -10
; RV64IM-NEXT:  rem a0, a1, a0
; RV64IM-NEXT:  ret
; %1 = srem i64 -10, %a
; ret i64 %1
}

```

```

define i8 @urem8(i8 %a, i8 %b) nounwind {
; RV64I-LABEL: urem8:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  andi a0, a0, 255
; RV64I-NEXT:  andi a1, a1, 255
; RV64I-NEXT:  call __umoddi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: urem8:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  andi a1, a1, 255
; RV64IM-NEXT:  andi a0, a0, 255
; RV64IM-NEXT:  remuw a0, a0, a1
; RV64IM-NEXT:  ret
; %1 = urem i8 %a, %b
; ret i8 %1
}

```

```

define i8 @urem8_constant_lhs(i8 %a) nounwind {
; RV64I-LABEL: urem8_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  andi a1, a0, 255
; RV64I-NEXT:

```

```

    li a0, 10
; RV64I-NEXT:  call __umoddi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: urem8_constant_lhs:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  andi a0, a0, 255
; RV64IM-NEXT:  li a1, 10
; RV64IM-NEXT:  remuw a0, a1, a0
; RV64IM-NEXT:  ret
%1 = urem i8 10, %a
ret i8 %1
}

```

```

define i8 @srem8(i8 %a, i8 %b) nounwind {
; RV64I-LABEL: srem8:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a1, a1, 24
; RV64I-NEXT:  sraiw a1, a1, 24
; RV64I-NEXT:  slli a0, a0, 24
; RV64I-NEXT:  sraiw a0, a0, 24
; RV64I-NEXT:  call __moddi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: srem8:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  slli a1, a1, 24
; RV64IM-NEXT:  sraiw a1, a1, 24
; RV64IM-NEXT:  slli a0, a0, 24
; RV64IM-NEXT:  sraiw
a0, a0, 24
; RV64IM-NEXT:  remw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = srem i8 %a, %b
ret i8 %1
}

```

```

define i8 @srem8_constant_lhs(i8 %a) nounwind {
; RV64I-LABEL: srem8_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16

```

```

; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT: slli a0, a0, 24
; RV64I-NEXT: sraiw a1, a0, 24
; RV64I-NEXT: li a0, -10
; RV64I-NEXT: call __moddi3
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT: addi sp, sp, 16
; RV64I-NEXT: ret
;
; RV64IM-LABEL: srem8_constant_lhs:
; RV64IM: # %bb.0:
; RV64IM-NEXT: slli a0, a0, 24
; RV64IM-NEXT: sraiw a0, a0, 24
; RV64IM-NEXT: li a1, -10
; RV64IM-NEXT: remw a0, a1, a0
; RV64IM-NEXT: ret
%1 = srem i8 -10, %a
ret i8 %1
}

```

```

define i16 @urem16(i16 %a, i16 %b) nounwind {
; RV64I-LABEL: urem16:
; RV64I: # %bb.0:
; RV64I-NEXT: addi sp, sp, -16
; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT: lui a2, 16
; RV64I-NEXT:
addiw a2, a2, -1
; RV64I-NEXT: and a0, a0, a2
; RV64I-NEXT: and a1, a1, a2
; RV64I-NEXT: call __umoddi3
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT: addi sp, sp, 16
; RV64I-NEXT: ret
;
; RV64IM-LABEL: urem16:
; RV64IM: # %bb.0:
; RV64IM-NEXT: lui a2, 16
; RV64IM-NEXT: addi a2, a2, -1
; RV64IM-NEXT: and a1, a1, a2
; RV64IM-NEXT: and a0, a0, a2
; RV64IM-NEXT: remuw a0, a0, a1
; RV64IM-NEXT: ret
%1 = urem i16 %a, %b
ret i16 %1
}

```



```

define i16 @urem16_constant_lhs(i16 %a) nounwind {
; RV64I-LABEL: urem16_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srli a1, a0, 48
; RV64I-NEXT:  li a0, 10
; RV64I-NEXT:  call __umoddi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: urem16_constant_lhs:
; RV64IM:      # %bb.0:
;
; RV64IM-NEXT:  slli a0, a0, 48
; RV64IM-NEXT:  srli a0, a0, 48
; RV64IM-NEXT:  li a1, 10
; RV64IM-NEXT:  remuw a0, a1, a0
; RV64IM-NEXT:  ret
%1 = urem i16 10, %a
ret i16 %1
}

```

```

define i16 @srem16(i16 %a, i16 %b) nounwind {
; RV64I-LABEL: srem16:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a1, a1, 16
; RV64I-NEXT:  sraiw a1, a1, 16
; RV64I-NEXT:  slli a0, a0, 16
; RV64I-NEXT:  sraiw a0, a0, 16
; RV64I-NEXT:  call __moddi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: srem16:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  slli a1, a1, 16
; RV64IM-NEXT:  sraiw a1, a1, 16
; RV64IM-NEXT:  slli a0, a0, 16
; RV64IM-NEXT:  sraiw a0, a0, 16
; RV64IM-NEXT:  remw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = srem i16 %a, %b

```

```

ret i16 %1
}

define i16 @srem16_constant_lhs(i16 %a) nounwind {
; RV64I-LABEL: srem16_constant_lhs:
;
RV64I:    # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a0, a0, 16
; RV64I-NEXT:  sraiw a1, a0, 16
; RV64I-NEXT:  li a0, -10
; RV64I-NEXT:  call __moddi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: srem16_constant_lhs:
; RV64IM:    # %bb.0:
; RV64IM-NEXT:  slli a0, a0, 16
; RV64IM-NEXT:  sraiw a0, a0, 16
; RV64IM-NEXT:  li a1, -10
; RV64IM-NEXT:  remw a0, a1, a0
; RV64IM-NEXT:  ret
%1 = srem i16 -10, %a
ret i16 %1
}

; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=aarch64-apple-ios %s -o - | FileCheck %s

```

```

define <16 x double> @test_sitofp_fixed(<16 x i32> %in) {
; CHECK-LABEL: test_sitofp_fixed:
; CHECK:      ; %bb.0:
; CHECK-NEXT:  sshll2.2d v4, v0, #0
; CHECK-NEXT:  sshll.2d v0, v0, #0
; CHECK-NEXT:  sshll2.2d v5, v1, #0
; CHECK-NEXT:  sshll.2d v6, v1, #0
; CHECK-NEXT:  sshll.2d v7, v2, #0
; CHECK-NEXT:  sshll2.2d v16, v2, #0
; CHECK-NEXT:  sshll2.2d v17, v3, #0
; CHECK-NEXT:  sshll.2d v18, v3, #0
; CHECK-NEXT:  scvtf.2d v1, v4, #6
; CHECK-NEXT:  scvtf.2d v0, v0, #6
; CHECK-NEXT:  scvtf.2d v3, v5, #6
; CHECK-NEXT:  scvtf.2d v2, v6, #6
; CHECK-NEXT:  scvtf.2d v4, v7, #6
; CHECK-NEXT:  scvtf.2d v5, v16, #6
; CHECK-NEXT:  scvtf.2d v7, v17, #6

```

```
; CHECK-NEXT:   scvtf.2d v6, v18, #6
; CHECK-NEXT:   ret

%flt = sitofp <16 x i32> %in to <16 x double>
%res = fdiv <16 x double> %flt, <double 64.0, double 64.0, double 64.0,
double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double
64.0, double 64.0, double 64.0, double 64.0, double 64.0>
ret <16 x double> %res
}
```

; This one is small enough to satisfy isSimple, but still illegally large.

```
define <4 x double> @test_sitofp_fixed_shortish(<4 x i64> %in) {
```

```
; CHECK-LABEL: test_sitofp_fixed_shortish:
```

```
; CHECK:      ; %bb.0:
```

```
; CHECK-NEXT:   scvtf.2d v0, v0, #6
```

```
; CHECK-NEXT:   scvtf.2d v1, v1, #6
```

```
; CHECK-NEXT:   ret
```

```
%flt = sitofp <4 x i64> %in to <4 x double>
```

```
%res = fdiv <4 x double> %flt, <double 64.0, double 64.0, double 64.0, double 64.0>
```

```
ret <4 x double> %res
```

```
}
```

; NOTE: Assertions have been autogenerated by utils/update\_llc\_test\_checks.py

```
; RUN: llc -mtriple=riscv64 -verify-machineinstrs < %s \
```

```
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64I
```

; These tests are each targeted at a particular RISC-V ALU instruction. Most

; other files in this folder exercise LLVM IR instructions that don't directly

; match a RISC-V instruction.

; Register-immediate instructions.

```
define i32 @addi(i32 %a) nounwind {
```

```
; RV64I-LABEL: addi:
```

```
; RV64I:      # %bb.0:
```

```
; RV64I-NEXT:   addiw a0, a0, 1
```

```
; RV64I-NEXT:   ret
```

```
%1 = add i32 %a, 1
```

```
ret i32 %1
```

```
}
```

```
define i32 @slti(i32 %a) nounwind {
```

```
; RV64I-LABEL: slti:
```

```
; RV64I:      # %bb.0:
```

```
; RV64I-NEXT:   sext.w a0, a0
```

```
; RV64I-NEXT:   slti a0, a0, 2
```

```
; RV64I-NEXT:   ret
```

```

%1 = icmp slt i32 %a, 2
%2 = zext i1 %1 to i32
ret i32 %2
}

```

```

define i32 @sltiu(i32 %a) nounwind {
; RV64I-LABEL: sltiu:
; RV64I:      # %bb.0:
; RV64I-NEXT: sext.w a0, a0
; RV64I-NEXT: sltiu a0, a0, 3
; RV64I-NEXT:
    ret
%1 = icmp ult i32 %a, 3
%2 = zext i1 %1 to i32
ret i32 %2
}

```

```

define i32 @xori(i32 %a) nounwind {
; RV64I-LABEL: xori:
; RV64I:      # %bb.0:
; RV64I-NEXT: xori a0, a0, 4
; RV64I-NEXT: ret
%1 = xor i32 %a, 4
ret i32 %1
}

```

```

define i32 @ori(i32 %a) nounwind {
; RV64I-LABEL: ori:
; RV64I:      # %bb.0:
; RV64I-NEXT: ori a0, a0, 5
; RV64I-NEXT: ret
%1 = or i32 %a, 5
ret i32 %1
}

```

```

define i32 @andi(i32 %a) nounwind {
; RV64I-LABEL: andi:
; RV64I:      # %bb.0:
; RV64I-NEXT: andi a0, a0, 6
; RV64I-NEXT: ret
%1 = and i32 %a, 6
ret i32 %1
}

```

```

define i32 @slli(i32 %a) nounwind {
; RV64I-LABEL: slli:
; RV64I:      # %bb.0:
; RV64I-NEXT: slliw a0, a0, 7

```

```
; RV64I-NEXT:  ret
%1 = shl i32 %a, 7
ret i32 %1
}
```

```
define i32 @srli(i32 %a) nounwind {
; RV64I-LABEL: srli:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srliw a0, a0, 8
; RV64I-NEXT:  ret
%1 = lshr i32 %a, 8
ret i32 %1
}
```

```
define i32 @srai(i32 %a) nounwind {
; RV64I-LABEL: srai:
; RV64I:
# %bb.0:
; RV64I-NEXT:  sraiw a0, a0, 9
; RV64I-NEXT:  ret
%1 = ashr i32 %a, 9
ret i32 %1
}
```

```
; Register-register instructions
```

```
define i32 @add(i32 %a, i32 %b) nounwind {
; RV64I-LABEL: add:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addw a0, a0, a1
; RV64I-NEXT:  ret
%1 = add i32 %a, %b
ret i32 %1
}
```

```
define i32 @sub(i32 %a, i32 %b) nounwind {
; RV64I-LABEL: sub:
; RV64I:      # %bb.0:
; RV64I-NEXT:  subw a0, a0, a1
; RV64I-NEXT:  ret
%1 = sub i32 %a, %b
ret i32 %1
}
```

```
define i32 @sub_negative_constant_lhs(i32 %a) nounwind {
; RV64I-LABEL: sub_negative_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, -2
```

```
; RV64I-NEXT:  subw a0, a1, a0
; RV64I-NEXT:  ret
%1 = sub i32 -2, %a
ret i32 %1
}
```

```
define i32 @sll(i32 %a, i32 %b) nounwind {
; RV64I-LABEL: sll:
; RV64I:      # %bb.0:
; RV64I-NEXT:  sllw a0, a0, a1
; RV64I-NEXT:  ret
%1 = shl i32 %a, %b
ret i32 %1
}
```

; Make sure we don't emit instructions to zero extend the shift amount to i64.

```
define
i32 @sll_shamt_zext(i32 %a, i32 %b) nounwind {
; RV64I-LABEL: sll_shamt_zext:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi a1, a1, 1
; RV64I-NEXT:  sllw a0, a0, a1
; RV64I-NEXT:  ret
%shamt = add i32 %b, 1
%1 = shl i32 %a, %shamt
ret i32 %1
}
```

```
define i32 @sll_negative_constant_lhs(i32 %a) nounwind {
; RV64I-LABEL: sll_negative_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, -1
; RV64I-NEXT:  sllw a0, a1, a0
; RV64I-NEXT:  ret
%1 = shl i32 -1, %a
ret i32 %1
}
```

```
define i32 @slt(i32 %a, i32 %b) nounwind {
; RV64I-LABEL: slt:
; RV64I:      # %bb.0:
; RV64I-NEXT:  sext.w a1, a1
; RV64I-NEXT:  sext.w a0, a0
; RV64I-NEXT:  slt a0, a0, a1
; RV64I-NEXT:  ret
%1 = icmp slt i32 %a, %b
%2 = zext i1 %1 to i32
ret i32 %2
}
```

```
}
```

```
define i32 @sltu(i32 %a, i32 %b) nounwind {  
;  
; RV64I-LABEL: sltu:  
; RV64I:      # %bb.0:  
; RV64I-NEXT: sext.w a1, a1  
; RV64I-NEXT: sext.w a0, a0  
; RV64I-NEXT: sltu a0, a0, a1  
; RV64I-NEXT: ret  
%1 = icmp ult i32 %a,  
%b  
%2 = zext i1 %1 to i32  
ret i32 %2  
}
```

```
define i32 @xor(i32 %a, i32 %b) nounwind {  
;  
; RV64I-LABEL: xor:  
; RV64I:      # %bb.0:  
; RV64I-NEXT: xor a0, a0, a1  
; RV64I-NEXT: ret  
%1 = xor i32 %a, %b  
ret i32 %1  
}
```

```
define i32 @srl(i32 %a, i32 %b) nounwind {  
;  
; RV64I-LABEL: srl:  
; RV64I:      # %bb.0:  
; RV64I-NEXT: srlw a0, a0, a1  
; RV64I-NEXT: ret  
%1 = lshr i32 %a, %b  
ret i32 %1  
}
```

; Make sure we don't emit instructions to zero extend the shift amount to i64.

```
define i32 @srl_shamt_zext(i32 %a, i32 %b) nounwind {  
; RV64I-LABEL: srl_shamt_zext:  
; RV64I:      # %bb.0:  
; RV64I-NEXT: addi a1, a1, 1  
; RV64I-NEXT: srlw a0, a0, a1  
; RV64I-NEXT: ret  
%shamt = add i32 %b, 1  
%1 = lshr i32 %a, %shamt  
ret i32 %1  
}
```

```

define i32 @srl_negative_constant_lhs(i32 %a) nounwind {
;
; RV64I-LABEL: srl_negative_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, -1
; RV64I-NEXT:  srlw a0, a1, a0
; RV64I-NEXT:  ret
%1 = lshr i32 -1, %a
ret i32 %1
}

define
i32 @sra(i32 %a, i32 %b) nounwind {
;
; RV64I-LABEL: sra:
; RV64I:      # %bb.0:
; RV64I-NEXT:  sraw a0, a0, a1
; RV64I-NEXT:  ret
%1 = ashr i32 %a, %b
ret i32 %1
}

; Make sure we don't emit instructions to zero extend the shift amount to i64.
define i32 @sra_shamt_zext(i32 %a, i32 %b) nounwind {
; RV64I-LABEL: sra_shamt_zext:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi a1, a1, 1
; RV64I-NEXT:  sraw a0, a0, a1
; RV64I-NEXT:  ret
%shamt = add i32 %b, 1
%1 = ashr i32 %a, %shamt
ret i32 %1
}

define i32 @sra_negative_constant_lhs(i32 %a) nounwind {
;
; RV64I-LABEL: sra_negative_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a1, 524288
; RV64I-NEXT:  sraw a0, a1, a0
; RV64I-NEXT:  ret
%1 = ashr i32 2147483648, %a
ret i32 %1
}

define i32 @or(i32 %a, i32 %b) nounwind {
;
; RV64I-LABEL: or:

```



```

; RV64I:    # %bb.0:
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
%1 = or i32 %a, %b
ret i32 %1
}

define i32 @and(i32 %a, i32 %b) nounwind {
;
; RV64I-LABEL:
and:
; RV64I:    # %bb.0:
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  ret
%1 = and i32 %a, %b
ret i32 %1
}
; NOTE: Assertions have been autogenerated by utils/update_test_checks.py UTC_ARGS: --include-generated-
funcs
; RUN: opt -S -passes=verify,iroutliner -ir-outlining-no-cost < %s | FileCheck %s

; This test checks that we do outline indirect calls when it is not specified
; that we should not.

declare void @f1(ptr, ptr);
declare void @f2(ptr, ptr);

define void @function1(ptr %func) {
entry:
%a = alloca i32, align 4
%b = alloca i32, align 4
%c = alloca i32, align 4
store i32 2, ptr %a, align 4
store i32 3, ptr %b, align 4
store i32 4, ptr %c, align 4
call void %func()
%al = load i32, ptr %a
%bl = load i32, ptr %b
%cl = load i32, ptr %c
ret void
}

define void @function2(ptr %func) {
entry:
%a = alloca i32, align 4
%b = alloca i32, align 4
%c = alloca i32, align 4
store i32 2, ptr %a, align 4

```

```

store i32 3, ptr %b, align 4
store i32 4, ptr %c, align 4
call void %func()
%al = load i32, ptr %a
%bl = load i32, ptr %b
%cl = load i32, ptr %c
ret void
}
;
CHECK-LABEL: @function1(
; CHECK-NEXT: entry:
; CHECK-NEXT: [[A:%.*]] = alloca i32, align 4
; CHECK-NEXT: [[B:%.*]] = alloca i32, align 4
; CHECK-NEXT: [[C:%.*]] = alloca i32, align 4
; CHECK-NEXT: call void @outlined_ir_func_0(ptr [[A]], ptr [[B]], ptr [[C]], ptr [[FUNC:%.*]])
; CHECK-NEXT: ret void
;
;
; CHECK-LABEL: @function2(
; CHECK-NEXT: entry:
; CHECK-NEXT: [[A:%.*]] = alloca i32, align 4
; CHECK-NEXT: [[B:%.*]] = alloca i32, align 4
; CHECK-NEXT: [[C:%.*]] = alloca i32, align 4
; CHECK-NEXT: call void @outlined_ir_func_0(ptr [[A]], ptr [[B]], ptr [[C]], ptr [[FUNC:%.*]])
; CHECK-NEXT: ret void
;
;
; CHECK-LABEL: define internal void @outlined_ir_func_0(
; CHECK-NEXT: newFuncRoot:
; CHECK-NEXT: br label [[ENTRY_TO_OUTLINE:%.*]]
; CHECK: entry_to_outline:
; CHECK-NEXT: store i32 2, ptr [[TMP0:%.*]], align 4
; CHECK-NEXT: store i32 3, ptr [[TMP1:%.*]], align 4
; CHECK-NEXT: store i32 4, ptr [[TMP2:%.*]], align 4
; CHECK-NEXT:
call void [[TMP3:%.*]]()
; CHECK-NEXT: [[AL:%.*]] = load i32, ptr [[TMP0]], align 4
; CHECK-NEXT: [[BL:%.*]] = load i32, ptr [[TMP1]], align 4
; CHECK-NEXT: [[CL:%.*]] = load i32, ptr [[TMP2]], align 4
; CHECK-NEXT: br label [[ENTRY_AFTER_OUTLINE_EXITSTUB:%.*]]
; CHECK: entry_after_outline.exitStub:
; CHECK-NEXT: ret void
;
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64I
; RUN: llc -mtriple=riscv64 -mattr=+xtheadbb -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64XTHEADBB

```

```

declare i32 @llvmctlz.i32(i32, i1)

define signext i32 @ctlz_i32(i32 signext %a) nounwind {
; RV64I-LABEL: ctlz_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT: beqz a0, .LBB0_2
; RV64I-NEXT: # %bb.1: # %cond.false
; RV64I-NEXT: addi sp, sp, -16
; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT: srliw a1, a0, 1
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 2
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 4
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 8
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 16
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT:
    not a0, a0
; RV64I-NEXT: srliw a1, a0, 1
; RV64I-NEXT: lui a2, 349525
; RV64I-NEXT: addi a2, a2, 1365
; RV64I-NEXT: and a1, a1, a2
; RV64I-NEXT: subw a0, a0, a1
; RV64I-NEXT: lui a1, 209715
; RV64I-NEXT: addi a1, a1, 819
; RV64I-NEXT: and a2, a0, a1
; RV64I-NEXT: srliw a0, a0, 2
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: add a0, a2, a0
; RV64I-NEXT: srliw a1, a0, 4
; RV64I-NEXT: addw a0, a0, a1
; RV64I-NEXT: lui a1, 61681
; RV64I-NEXT: addiw a1, a1, -241
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: lui a1, 4112
; RV64I-NEXT: addiw a1, a1, 257
; RV64I-NEXT: call __muldi3
; RV64I-NEXT: srliw a0, a0, 24
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT: addi sp, sp, 16
; RV64I-NEXT: ret
; RV64I-NEXT: .LBB0_2:
; RV64I-NEXT: li a0, 32
; RV64I-NEXT: ret

```

```

;
; RV64XTHEADBB-LABEL: ctlz_i32:
; RV64XTHEADBB:    # %bb.0:
; RV64XTHEADBB-NEXT:  th.extu a0, a0, 31, 0
; RV64XTHEADBB-NEXT:  th.ff1 a0,
a0
; RV64XTHEADBB-NEXT:  addi a0, a0, -32
; RV64XTHEADBB-NEXT:  ret
%1 = call i32 @llvm.ctlz.i32(i32 %a, i1 false)
ret i32 %1
}

```

```

define signext i32 @log2_i32(i32 signext %a) nounwind {
; RV64I-LABEL: log2_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT:  beqz a0, .LBB1_2
; RV64I-NEXT:  # %bb.1: # %cond.false
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  srliw a1, a0, 1
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 2
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 4
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 8
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 16
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  srliw a1, a0, 1
; RV64I-NEXT:  lui a2, 349525
; RV64I-NEXT:  addi a2, a2, 1365
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  subw a0, a0, a1
; RV64I-NEXT:  lui a1, 209715
; RV64I-NEXT:  addi a1, a1, 819
; RV64I-NEXT:
and a2, a0, a1
; RV64I-NEXT:  srliw a0, a0, 2
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  add a0, a2, a0
; RV64I-NEXT:  srliw a1, a0, 4
; RV64I-NEXT:  addw a0, a0, a1
; RV64I-NEXT:  lui a1, 61681
; RV64I-NEXT:  addiw a1, a1, -241
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  lui a1, 4112

```

```

; RV64I-NEXT:    addiw a1, a1, 257
; RV64I-NEXT:    call __muldi3
; RV64I-NEXT:    srliw a0, a0, 24
; RV64I-NEXT:    ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:    addi sp, sp, 16
; RV64I-NEXT:    j .LBB1_3
; RV64I-NEXT:    .LBB1_2:
; RV64I-NEXT:    li a0, 32
; RV64I-NEXT:    .LBB1_3: # %cond.end
; RV64I-NEXT:    li a1, 31
; RV64I-NEXT:    subw a0, a1, a0
; RV64I-NEXT:    ret
;
; RV64XTHEADBB-LABEL: log2_i32:
; RV64XTHEADBB:    # %bb.0:
; RV64XTHEADBB-NEXT:  th.extu a0, a0, 31, 0
; RV64XTHEADBB-NEXT:  th.ff1 a0, a0
; RV64XTHEADBB-NEXT:  addi a0, a0, -32
; RV64XTHEADBB-NEXT:  li a1, 31
; RV64XTHEADBB-NEXT:  subw a0, a1, a0
; RV64XTHEADBB-NEXT:  ret
%1 = call i32 @llvm.ctlz.i32(i32
%a, i1 false)
%2 = sub i32 31, %1
ret i32 %2
}

define signext i32 @log2_ceil_i32(i32 signext %a) nounwind {
; RV64I-LABEL: log2_ceil_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  sd s0, 0(sp) # 8-byte Folded Spill
; RV64I-NEXT:  addiw a0, a0, -1
; RV64I-NEXT:  li s0, 32
; RV64I-NEXT:  li a1, 32
; RV64I-NEXT:  beqz a0, .LBB2_2
; RV64I-NEXT:  # %bb.1: # %cond.false
; RV64I-NEXT:  srliw a1, a0, 1
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 2
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 4
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 8
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 16
; RV64I-NEXT:  or a0, a0, a1

```

```

; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  srlw a1, a0, 1
; RV64I-NEXT:  lui a2, 349525
; RV64I-NEXT:  addi a2, a2, 1365
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  subw a0, a0, a1
; RV64I-NEXT:
    lui a1, 209715
; RV64I-NEXT:  addi a1, a1, 819
; RV64I-NEXT:  and a2, a0, a1
; RV64I-NEXT:  srlw a0, a0, 2
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  add a0, a2, a0
; RV64I-NEXT:  srlw a1, a0, 4
; RV64I-NEXT:  addw a0, a0, a1
; RV64I-NEXT:  lui a1, 61681
; RV64I-NEXT:  addiw a1, a1, -241
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  lui a1, 4112
; RV64I-NEXT:  addiw a1, a1, 257
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:  srlw a1, a0, 24
; RV64I-NEXT:  .LBB2_2: # %cond.end
; RV64I-NEXT:  subw a0, s0, a1
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  ld s0, 0(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: log2_ceil_i32:
; RV64XTHEADBB:    # %bb.0:
; RV64XTHEADBB-NEXT:  addi a0, a0, -1
; RV64XTHEADBB-NEXT:  slli a0, a0, 32
; RV64XTHEADBB-NEXT:  srli a0, a0, 32
; RV64XTHEADBB-NEXT:  th.ff1 a0, a0
; RV64XTHEADBB-NEXT:  addi a0, a0, -32
; RV64XTHEADBB-NEXT:
    li a1, 32
; RV64XTHEADBB-NEXT:  subw a0, a1, a0
; RV64XTHEADBB-NEXT:  ret
%1 = sub i32 %a, 1
%2 = call i32 @llvmctlz.i32(i32 %1, i1 false)
%3 = sub i32 32, %2
ret i32 %3
}

define signext i32 @findLastSet_i32(i32 signext %a) nounwind {
; RV64I-LABEL: findLastSet_i32:

```

```

; RV64I:    # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  sd s0, 0(sp) # 8-byte Folded Spill
; RV64I-NEXT:  mv s0, a0
; RV64I-NEXT:  srliw a0, a0, 1
; RV64I-NEXT:  or a0, s0, a0
; RV64I-NEXT:  srliw a1, a0, 2
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 4
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 8
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 16
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  srliw a1, a0, 1
; RV64I-NEXT:  lui a2, 349525
; RV64I-NEXT:  addi a2, a2, 1365
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  subw a0, a0, a1
; RV64I-NEXT:
    lui a1, 209715
; RV64I-NEXT:  addi a1, a1, 819
; RV64I-NEXT:  and a2, a0, a1
; RV64I-NEXT:  srliw a0, a0, 2
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  add a0, a2, a0
; RV64I-NEXT:  srliw a1, a0, 4
; RV64I-NEXT:  addw a0, a0, a1
; RV64I-NEXT:  lui a1, 61681
; RV64I-NEXT:  addiw a1, a1, -241
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  lui a1, 4112
; RV64I-NEXT:  addiw a1, a1, 257
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:  srliw a0, a0, 24
; RV64I-NEXT:  xori a0, a0, 31
; RV64I-NEXT:  snez a1, s0
; RV64I-NEXT:  addiw a1, a1, -1
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  ld s0, 0(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: findLastSet_i32:
; RV64XTHEADBB:    # %bb.0:

```

```

; RV64XTHEADBB-NEXT: th.extu a1, a0, 31, 0
; RV64XTHEADBB-NEXT: th.ff1 a1, a1
; RV64XTHEADBB-NEXT: addiw a1, a1, -32
; RV64XTHEADBB-NEXT: xori a1, a1,
31
; RV64XTHEADBB-NEXT: snez a0, a0
; RV64XTHEADBB-NEXT: addiw a0, a0, -1
; RV64XTHEADBB-NEXT: or a0, a0, a1
; RV64XTHEADBB-NEXT: ret
%1 = call i32 @llvm.ctlz.i32(i32 %a, i1 true)
%2 = xor i32 31, %1
%3 = icmp eq i32 %a, 0
%4 = select i1 %3, i32 -1, i32 %2
ret i32 %4
}

```

```

define i32 @ctlz_lshr_i32(i32 signext %a) {
; RV64I-LABEL: ctlz_lshr_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT: srliw a0, a0, 1
; RV64I-NEXT: beqz a0, .LBB4_2
; RV64I-NEXT: # %bb.1: # %cond.false
; RV64I-NEXT: addi sp, sp, -16
; RV64I-NEXT: .cfi_def_cfa_offset 16
; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT: .cfi_offset ra, -8
; RV64I-NEXT: srliw a1, a0, 1
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 2
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 4
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 8
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 16
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: not
a0, a0
; RV64I-NEXT: srliw a1, a0, 1
; RV64I-NEXT: lui a2, 349525
; RV64I-NEXT: addi a2, a2, 1365
; RV64I-NEXT: and a1, a1, a2
; RV64I-NEXT: subw a0, a0, a1
; RV64I-NEXT: lui a1, 209715
; RV64I-NEXT: addi a1, a1, 819
; RV64I-NEXT: and a2, a0, a1
; RV64I-NEXT: srliw a0, a0, 2
; RV64I-NEXT: and a0, a0, a1

```



```

; RV64I-NEXT:  add a0, a2, a0
; RV64I-NEXT:  srliw a1, a0, 4
; RV64I-NEXT:  addw a0, a0, a1
; RV64I-NEXT:  lui a1, 61681
; RV64I-NEXT:  addiw a1, a1, -241
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  lui a1, 4112
; RV64I-NEXT:  addiw a1, a1, 257
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:  srliw a0, a0, 24
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
; RV64I-NEXT:  .LBB4_2:
; RV64I-NEXT:  li a0, 32
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: ctlz_lshr_i32:
; RV64XTHEADBB:    # %bb.0:
; RV64XTHEADBB-NEXT:  srliw a0, a0, 1
; RV64XTHEADBB-NEXT:  th.ff1 a0, a0
; RV64XTHEADBB-NEXT:
    addi a0, a0, -32
; RV64XTHEADBB-NEXT:  ret
%1 = lshr i32 %a, 1
%2 = call i32 @llvm.ctlz.i32(i32 %1, i1 false)
ret i32 %2
}

```

```
declare i64 @llvm.ctlz.i64(i64, i1)
```

```

define i64 @ctlz_i64(i64 %a) nounwind {
; RV64I-LABEL: ctlz_i64:
; RV64I:    # %bb.0:
; RV64I-NEXT:  beqz a0, .LBB5_2
; RV64I-NEXT:  # %bb.1: # %cond.false
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  srli a1, a0, 1
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srli a1, a0, 2
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srli a1, a0, 4
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srli a1, a0, 8
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srli a1, a0, 16
; RV64I-NEXT:  or a0, a0, a1

```

```

; RV64I-NEXT: srli a1, a0, 32
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: not a0, a0
; RV64I-NEXT: srli a1, a0, 1
; RV64I-NEXT: lui a2, 349525
; RV64I-NEXT: addiw a2, a2, 1365
; RV64I-NEXT: slli a3, a2, 32
; RV64I-NEXT: add a2, a2,
a3
; RV64I-NEXT: and a1, a1, a2
; RV64I-NEXT: sub a0, a0, a1
; RV64I-NEXT: lui a1, 209715
; RV64I-NEXT: addiw a1, a1, 819
; RV64I-NEXT: slli a2, a1, 32
; RV64I-NEXT: add a1, a1, a2
; RV64I-NEXT: and a2, a0, a1
; RV64I-NEXT: srli a0, a0, 2
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: add a0, a2, a0
; RV64I-NEXT: srli a1, a0, 4
; RV64I-NEXT: add a0, a0, a1
; RV64I-NEXT: lui a1, 61681
; RV64I-NEXT: addiw a1, a1, -241
; RV64I-NEXT: slli a2, a1, 32
; RV64I-NEXT: add a1, a1, a2
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: lui a1, 4112
; RV64I-NEXT: addiw a1, a1, 257
; RV64I-NEXT: slli a2, a1, 32
; RV64I-NEXT: add a1, a1, a2
; RV64I-NEXT: call __muldi3
; RV64I-NEXT: srli a0, a0, 56
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT: addi sp, sp, 16
; RV64I-NEXT: ret
; RV64I-NEXT: .LBB5_2:
; RV64I-NEXT: li a0, 64
; RV64I-NEXT: ret
;
; RV64XTHEADBB-LABEL: ctlz_i64:
; RV64XTHEADBB:    # %bb.0:
; RV64XTHEADBB-NEXT:
    th.ff1 a0, a0
; RV64XTHEADBB-NEXT: ret
%1 = call i64 @llvm.ctlz.i64(i64 %a, i1 false)
ret i64 %1
}

```

```
declare i32 @llvm.cttz.i32(i32, i1)
```

```
define signext i32 @cttz_i32(i32 signext %a) nounwind {  
; RV64I-LABEL: cttz_i32:  
; RV64I:      # %bb.0:  
; RV64I-NEXT: beqz a0, .LBB6_2  
; RV64I-NEXT: # %bb.1: # %cond.false  
; RV64I-NEXT: addi sp, sp, -16  
; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill  
; RV64I-NEXT: negw a1, a0  
; RV64I-NEXT: and a0, a0, a1  
; RV64I-NEXT: lui a1, 30667  
; RV64I-NEXT: addiw a1, a1, 1329  
; RV64I-NEXT: call __muldi3  
; RV64I-NEXT: srliw a0, a0, 27  
; RV64I-NEXT: lui a1, %hi(.LCPI6_0)  
; RV64I-NEXT: addi a1, a1, %lo(.LCPI6_0)  
; RV64I-NEXT: add a0, a1, a0  
; RV64I-NEXT: lbu a0, 0(a0)  
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload  
; RV64I-NEXT: addi sp, sp, 16  
; RV64I-NEXT: ret  
; RV64I-NEXT: .LBB6_2:  
; RV64I-NEXT: li a0, 32  
; RV64I-NEXT: ret  
;  
; RV64XTHEADBB-LABEL: cttz_i32:  
; RV64XTHEADBB:      #  
%bb.0:  
; RV64XTHEADBB-NEXT: beqz a0, .LBB6_2  
; RV64XTHEADBB-NEXT: # %bb.1: # %cond.false  
; RV64XTHEADBB-NEXT: addi sp, sp, -16  
; RV64XTHEADBB-NEXT: sd ra, 8(sp) # 8-byte Folded Spill  
; RV64XTHEADBB-NEXT: negw a1, a0  
; RV64XTHEADBB-NEXT: and a0, a0, a1  
; RV64XTHEADBB-NEXT: lui a1, 30667  
; RV64XTHEADBB-NEXT: addiw a1, a1, 1329  
; RV64XTHEADBB-NEXT: call __muldi3  
; RV64XTHEADBB-NEXT: srliw a0, a0, 27  
; RV64XTHEADBB-NEXT: lui a1, %hi(.LCPI6_0)  
; RV64XTHEADBB-NEXT: addi a1, a1, %lo(.LCPI6_0)  
; RV64XTHEADBB-NEXT: add a0, a1, a0  
; RV64XTHEADBB-NEXT: lbu a0, 0(a0)  
; RV64XTHEADBB-NEXT: ld ra, 8(sp) # 8-byte Folded Reload  
; RV64XTHEADBB-NEXT: addi sp, sp, 16  
; RV64XTHEADBB-NEXT: ret  
; RV64XTHEADBB-NEXT: .LBB6_2:  
; RV64XTHEADBB-NEXT: li a0, 32
```

```

; RV64XTHEADBB-NEXT:  ret
; RV64ZBB-LABEL: cttz_i32:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT:  ctw a0, a0
; RV64ZBB-NEXT:  ret
%1 = call i32 @llvm.cttz.i32(i32 %a, i1 false)
ret i32 %1
}

define
  signext i32 @cttz_zero_undef_i32(i32 signext %a) nounwind {
; RV64I-LABEL: cttz_zero_undef_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  negw a1, a0
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  lui a1, 30667
; RV64I-NEXT:  addiw a1, a1, 1329
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:  srliw a0, a0, 27
; RV64I-NEXT:  lui a1, %hi(.LCPI7_0)
; RV64I-NEXT:  addi a1, a1, %lo(.LCPI7_0)
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  lbu a0, 0(a0)
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: cttz_zero_undef_i32:
; RV64XTHEADBB:    # %bb.0:
; RV64XTHEADBB-NEXT:  addi sp, sp, -16
; RV64XTHEADBB-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64XTHEADBB-NEXT:  negw a1, a0
; RV64XTHEADBB-NEXT:  and a0, a0, a1
; RV64XTHEADBB-NEXT:  lui a1, 30667
; RV64XTHEADBB-NEXT:  addiw a1, a1, 1329
; RV64XTHEADBB-NEXT:
  call __muldi3
; RV64XTHEADBB-NEXT:  srliw a0, a0, 27
; RV64XTHEADBB-NEXT:  lui a1, %hi(.LCPI7_0)
; RV64XTHEADBB-NEXT:  addi a1, a1, %lo(.LCPI7_0)
; RV64XTHEADBB-NEXT:  add a0, a1, a0
; RV64XTHEADBB-NEXT:  lbu a0, 0(a0)
; RV64XTHEADBB-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64XTHEADBB-NEXT:  addi sp, sp, 16
; RV64XTHEADBB-NEXT:  ret
%1 = call i32 @llvm.cttz.i32(i32 %a, i1 true)

```

```

ret i32 %1
}

define signext i32 @findFirstSet_i32(i32 signext %a) nounwind {
; RV64I-LABEL: findFirstSet_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT: addi sp, sp, -16
; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT: sd s0, 0(sp) # 8-byte Folded Spill
; RV64I-NEXT: mv s0, a0
; RV64I-NEXT: negw a0, a0
; RV64I-NEXT: and a0, s0, a0
; RV64I-NEXT: lui a1, 30667
; RV64I-NEXT: addiw a1, a1, 1329
; RV64I-NEXT: call __muldi3
; RV64I-NEXT: srliw a0, a0, 27
; RV64I-NEXT: lui a1, %hi(.LCPI8_0)
; RV64I-NEXT: addi a1, a1, %lo(.LCPI8_0)
; RV64I-NEXT:
    add a0, a1, a0
; RV64I-NEXT: lbu a0, 0(a0)
; RV64I-NEXT: snez a1, s0
; RV64I-NEXT: addi a1, a1, -1
; RV64I-NEXT: or a0, a1, a0
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT: ld s0, 0(sp) # 8-byte Folded Reload
; RV64I-NEXT: addi sp, sp, 16
; RV64I-NEXT: ret
;
; RV64XTHEADBB-LABEL: findFirstSet_i32:
; RV64XTHEADBB:      # %bb.0:
; RV64XTHEADBB-NEXT: addi sp, sp, -16
; RV64XTHEADBB-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64XTHEADBB-NEXT: sd s0, 0(sp) # 8-byte Folded Spill
; RV64XTHEADBB-NEXT: mv s0, a0
; RV64XTHEADBB-NEXT: negw a0, a0
; RV64XTHEADBB-NEXT: and a0, s0, a0
; RV64XTHEADBB-NEXT: lui a1, 30667
; RV64XTHEADBB-NEXT: addiw a1, a1, 1329
; RV64XTHEADBB-NEXT: call __muldi3
; RV64XTHEADBB-NEXT: srliw a0, a0, 27
; RV64XTHEADBB-NEXT: lui a1, %hi(.LCPI8_0)
; RV64XTHEADBB-NEXT: addi a1, a1, %lo(.LCPI8_0)
; RV64XTHEADBB-NEXT: add a0, a1, a0
; RV64XTHEADBB-NEXT: lbu a0, 0(a0)
; RV64XTHEADBB-NEXT:
    snez a1, s0
; RV64XTHEADBB-NEXT: addi a1, a1, -1

```

```

; RV64XTHEADBB-NEXT:  or a0, a1, a0
; RV64XTHEADBB-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64XTHEADBB-NEXT:  ld s0, 0(sp) # 8-byte Folded Reload
; RV64XTHEADBB-NEXT:  addi sp, sp, 16
; RV64XTHEADBB-NEXT:  ret
%1 = call i32 @llvm.cttz.i32(i32 %a, i1 true)
%2 = icmp eq i32 %a, 0
%3 = select i1 %2, i32 -1, i32 %1
ret i32 %3
}

```

```

define signext i32 @ffs_i32(i32 signext %a) nounwind {

```

```

; RV64I-LABEL: ffs_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  sd s0, 0(sp) # 8-byte Folded Spill
; RV64I-NEXT:  mv s0, a0
; RV64I-NEXT:  negw a0, a0
; RV64I-NEXT:  and a0, s0, a0
; RV64I-NEXT:  lui a1, 30667
; RV64I-NEXT:  addiw a1, a1, 1329
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:  srlw a0, a0, 27
; RV64I-NEXT:  lui a1, %hi(.LCPI9_0)
; RV64I-NEXT:  addi a1, a1, %lo(.LCPI9_0)
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:
    lbu a0, 0(a0)
; RV64I-NEXT:  addi a0, a0, 1
; RV64I-NEXT:  seqz a1, s0
; RV64I-NEXT:  addi a1, a1, -1
; RV64I-NEXT:  and a0, a1, a0
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  ld s0, 0(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: ffs_i32:
; RV64XTHEADBB:      # %bb.0:
; RV64XTHEADBB-NEXT:  addi sp, sp, -16
; RV64XTHEADBB-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64XTHEADBB-NEXT:  sd s0, 0(sp) # 8-byte Folded Spill
; RV64XTHEADBB-NEXT:  mv s0, a0
; RV64XTHEADBB-NEXT:  negw a0, a0
; RV64XTHEADBB-NEXT:  and a0, s0, a0

```

```

; RV64XTHEADBB-NEXT: lui a1, 30667
; RV64XTHEADBB-NEXT: addiw a1, a1, 1329
; RV64XTHEADBB-NEXT: call __muldi3
; RV64XTHEADBB-NEXT: srliw a0, a0, 27
; RV64XTHEADBB-NEXT: lui a1, %hi(.LCPI9_0)
; RV64XTHEADBB-NEXT: addi a1, a1, %lo(.LCPI9_0)
; RV64XTHEADBB-NEXT: add a0, a1, a0
;
RV64XTHEADBB-NEXT: lbu a0, 0(a0)
; RV64XTHEADBB-NEXT: addi a0, a0, 1
; RV64XTHEADBB-NEXT: seqz a1, s0
; RV64XTHEADBB-NEXT: addi a1, a1, -1
; RV64XTHEADBB-NEXT: and a0, a1, a0
; RV64XTHEADBB-NEXT: slli a0, a0, 32
; RV64XTHEADBB-NEXT: srli a0, a0, 32
; RV64XTHEADBB-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64XTHEADBB-NEXT: ld s0, 0(sp) # 8-byte Folded Reload
; RV64XTHEADBB-NEXT: addi sp, sp, 16
; RV64XTHEADBB-NEXT: ret
%1 = call i32 @llvm.cttz.i32(i32 %a, i1 true)
%2 = add i32 %1, 1
%3 = icmp eq i32 %a, 0
%4 = select i1 %3, i32 0, i32 %2
ret i32 %4
}

```

```
declare i64 @llvm.cttz.i64(i64, i1)
```

```

define i64 @cttz_i64(i64 %a) nounwind {
; RV64I-LABEL: cttz_i64:
; RV64I:    # %bb.0:
; RV64I-NEXT: beqz a0, .LBB10_2
; RV64I-NEXT: # %bb.1: # %cond.false
; RV64I-NEXT: addi sp, sp, -16
; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT: neg a1, a0
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: lui a1, %hi(.LCPI10_0)
; RV64I-NEXT:
    ld a1, %lo(.LCPI10_0)(a1)
; RV64I-NEXT: call __muldi3
; RV64I-NEXT: srli a0, a0, 58
; RV64I-NEXT: lui a1, %hi(.LCPI10_1)
; RV64I-NEXT: addi a1, a1, %lo(.LCPI10_1)
; RV64I-NEXT: add a0, a1, a0
; RV64I-NEXT: lbu a0, 0(a0)
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT: addi sp, sp, 16

```

```

; RV64I-NEXT:  ret
; RV64I-NEXT:  .LBB10_2:
; RV64I-NEXT:  li a0, 64
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: cttz_i64:
; RV64XTHEADBB:      # %bb.0:
; RV64XTHEADBB-NEXT:  beqz a0, .LBB10_2
; RV64XTHEADBB-NEXT:  # %bb.1: # %cond.false
; RV64XTHEADBB-NEXT:  addi a1, a0, -1
; RV64XTHEADBB-NEXT:  not a0, a0
; RV64XTHEADBB-NEXT:  and a0, a0, a1
; RV64XTHEADBB-NEXT:  th.ff1 a0, a0
; RV64XTHEADBB-NEXT:  li a1, 64
; RV64XTHEADBB-NEXT:  sub a0, a1, a0
; RV64XTHEADBB-NEXT:  ret
; RV64XTHEADBB-NEXT: .LBB10_2:
; RV64XTHEADBB-NEXT:  li a0, 64
; RV64XTHEADBB-NEXT:  ret
%1 = call i64 @llvm.cttz.i64(i64 %a, i1 false)
ret i64 %1
}

```

```

define signext
i32 @sextb_i32(i32 signext %a) nounwind {
; RV64I-LABEL: sextb_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 56
; RV64I-NEXT:  srai a0, a0, 56
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: sextb_i32:
; RV64XTHEADBB:      # %bb.0:
; RV64XTHEADBB-NEXT:  th.ext a0, a0, 7, 0
; RV64XTHEADBB-NEXT:  ret
%shl = shl i32 %a, 24
%shr = ashr exact i32 %shl, 24
ret i32 %shr
}

```

```

define i64 @sextb_i64(i64 %a) nounwind {
; RV64I-LABEL: sextb_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 56
; RV64I-NEXT:  srai a0, a0, 56
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: sextb_i64:

```



```

; RV64XTHEADBB:    # %bb.0:
; RV64XTHEADBB-NEXT:  th.ext a0, a0, 7, 0
; RV64XTHEADBB-NEXT:  ret
%shl = shl i64 %a, 56
%shr = ashr exact i64 %shl, 56
ret i64 %shr
}

define signext i32 @sexth_i32(i32 signext %a) nounwind {
; RV64I-LABEL: sexth_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srai a0, a0, 48
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL:
sexth_i32:
; RV64XTHEADBB:    # %bb.0:
; RV64XTHEADBB-NEXT:  th.ext a0, a0, 15, 0
; RV64XTHEADBB-NEXT:  ret
%shl = shl i32 %a, 16
%shr = ashr exact i32 %shl, 16
ret i32 %shr
}

define signext i32 @no_sexth_i32(i32 signext %a) nounwind {
; RV64I-LABEL: no_sexth_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 17
; RV64I-NEXT:  sraiw a0, a0, 16
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: no_sexth_i32:
; RV64XTHEADBB:    # %bb.0:
; RV64XTHEADBB-NEXT:  slli a0, a0, 17
; RV64XTHEADBB-NEXT:  sraiw a0, a0, 16
; RV64XTHEADBB-NEXT:  ret
%shl = shl i32 %a, 17
%shr = ashr exact i32 %shl, 16
ret i32 %shr
}

define i64 @sexth_i64(i64 %a) nounwind {
; RV64I-LABEL: sexth_i64:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srai a0, a0, 48
; RV64I-NEXT:  ret

```

```

;
; RV64XTHEADBB-LABEL: sexth_i64:
; RV64XTHEADBB:    # %bb.0:
; RV64XTHEADBB-NEXT:  th.ext a0, a0, 15, 0
; RV64XTHEADBB-NEXT:  ret
%shl = shl i64 %a, 48
%shr = ashr
exact i64 %shl, 48
ret i64 %shr
}

define i64 @no_sexth_i64(i64 %a) nounwind {
; RV64I-LABEL: no_sexth_i64:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 49
; RV64I-NEXT:  srli a0, a0, 48
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: no_sexth_i64:
; RV64XTHEADBB:    # %bb.0:
; RV64XTHEADBB-NEXT:  slli a0, a0, 49
; RV64XTHEADBB-NEXT:  srli a0, a0, 48
; RV64XTHEADBB-NEXT:  ret
%shl = shl i64 %a, 49
%shr = ashr exact i64 %shl, 48
ret i64 %shr
}

define i32 @zexth_i32(i32 %a) nounwind {
; RV64I-LABEL: zexth_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srli a0, a0, 48
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: zexth_i32:
; RV64XTHEADBB:    # %bb.0:
; RV64XTHEADBB-NEXT:  th.extu a0, a0, 15, 0
; RV64XTHEADBB-NEXT:  ret
%and = and i32 %a, 65535
ret i32 %and
}

define i64 @zexth_i64(i64 %a) nounwind {
; RV64I-LABEL: zexth_i64:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srli a0, a0, 48

```

```

; RV64I-NEXT:
    ret
;
; RV64XTHEADBB-LABEL: zexth_i64:
; RV64XTHEADBB:    # %bb.0:
; RV64XTHEADBB-NEXT:  th.extu a0, a0, 15, 0
; RV64XTHEADBB-NEXT:  ret
    %and = and i64 %a, 65535
    ret i64 %and
}

```

```

define i64 @zext_bf_i64(i64 %a) nounwind {
; RV64I-LABEL: zext_bf_i64:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 47
; RV64I-NEXT:  srli a0, a0, 48
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: zext_bf_i64:
; RV64XTHEADBB:    # %bb.0:
; RV64XTHEADBB-NEXT:  th.extu a0, a0, 16, 1
; RV64XTHEADBB-NEXT:  ret
    %1 = lshr i64 %a, 1
    %and = and i64 %1, 65535
    ret i64 %and
}

```

```

define i64 @zext_i64_srliw(i64 %a) nounwind {
; RV64I-LABEL: zext_i64_srliw:
; RV64I:    # %bb.0:
; RV64I-NEXT:  srliw a0, a0, 16
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: zext_i64_srliw:
; RV64XTHEADBB:    # %bb.0:
; RV64XTHEADBB-NEXT:  srliw a0, a0, 16
; RV64XTHEADBB-NEXT:  ret
    %1 = lshr i64 %a, 16
    %and = and i64 %1, 65535
    ret i64 %and
}

```

```

declare i32 @llvm.bswap.i32(i32)

```

```

define signext
    i32 @bswap_i32(i32 signext %a) nounwind {
; RV64I-LABEL: bswap_i32:
; RV64I:    # %bb.0:

```

```

; RV64I-NEXT:  srlw a1, a0, 8
; RV64I-NEXT:  lui a2, 16
; RV64I-NEXT:  addiw a2, a2, -256
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  srlw a3, a0, 24
; RV64I-NEXT:  or a1, a1, a3
; RV64I-NEXT:  and a2, a0, a2
; RV64I-NEXT:  sllw a2, a2, 8
; RV64I-NEXT:  sllw a0, a0, 24
; RV64I-NEXT:  or a0, a0, a2
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: bswap_i32:
; RV64XTHEADBB:      # %bb.0:
; RV64XTHEADBB-NEXT:  th.revw a0, a0
; RV64XTHEADBB-NEXT:  ret
%1 = tail call i32 @llvm.bswap.i32(i32 %a)
ret i32 %1
}

```

; Similar to bswap\_i32 but the result is not sign extended.

```
define void @bswap_i32_nosext(i32 signext %a, ptr %x) nounwind {
```

```

; RV64I-LABEL: bswap_i32_nosext:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srlw a2, a0, 8
; RV64I-NEXT:  lui a3, 16
; RV64I-NEXT:  addi a3, a3, -256
; RV64I-NEXT:  and a2, a2, a3
; RV64I-NEXT:  srlw
a4, a0, 24
; RV64I-NEXT:  or a2, a2, a4
; RV64I-NEXT:  and a3, a0, a3
; RV64I-NEXT:  slli a3, a3, 8
; RV64I-NEXT:  slli a0, a0, 24
; RV64I-NEXT:  or a0, a0, a3
; RV64I-NEXT:  or a0, a0, a2
; RV64I-NEXT:  sw a0, 0(a1)
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: bswap_i32_nosext:
; RV64XTHEADBB:      # %bb.0:
; RV64XTHEADBB-NEXT:  th.revw a0, a0
; RV64XTHEADBB-NEXT:  sw a0, 0(a1)
; RV64XTHEADBB-NEXT:  ret
%1 = tail call i32 @llvm.bswap.i32(i32 %a)
store i32 %1, ptr %x
ret void

```

```
}
```

```
declare i64 @llvm.bswap.i64(i64)
```

```
define i64 @bswap_i64(i64 %a) {
```

```
; RV64I-LABEL: bswap_i64:
```

```
; RV64I:    # %bb.0:
```

```
; RV64I-NEXT:  srli a1, a0, 40
```

```
; RV64I-NEXT:  lui a2, 16
```

```
; RV64I-NEXT:  addiw a2, a2, -256
```

```
; RV64I-NEXT:  and a1, a1, a2
```

```
; RV64I-NEXT:  srli a3, a0, 56
```

```
; RV64I-NEXT:  or a1, a1, a3
```

```
; RV64I-NEXT:  srli a3, a0, 24
```

```
; RV64I-NEXT:  lui a4, 4080
```

```
; RV64I-NEXT:  and a3, a3, a4
```

```
; RV64I-NEXT:  srli a5, a0, 8
```

```
; RV64I-NEXT:  srliw a5, a5, 24
```

```
; RV64I-NEXT:  slli
```

```
    a5, a5, 24
```

```
; RV64I-NEXT:  or a3, a5, a3
```

```
; RV64I-NEXT:  or a1, a3, a1
```

```
; RV64I-NEXT:  and a4, a0, a4
```

```
; RV64I-NEXT:  slli a4, a4, 24
```

```
; RV64I-NEXT:  srliw a3, a0, 24
```

```
; RV64I-NEXT:  slli a3, a3, 32
```

```
; RV64I-NEXT:  or a3, a4, a3
```

```
; RV64I-NEXT:  and a2, a0, a2
```

```
; RV64I-NEXT:  slli a2, a2, 40
```

```
; RV64I-NEXT:  slli a0, a0, 56
```

```
; RV64I-NEXT:  or a0, a0, a2
```

```
; RV64I-NEXT:  or a0, a0, a3
```

```
; RV64I-NEXT:  or a0, a0, a1
```

```
; RV64I-NEXT:  ret
```

```
;
```

```
; RV64XTHEADBB-LABEL: bswap_i64:
```

```
; RV64XTHEADBB:    # %bb.0:
```

```
; RV64XTHEADBB-NEXT:  th.rev a0, a0
```

```
; RV64XTHEADBB-NEXT:  ret
```

```
    %1 = call i64 @llvm.bswap.i64(i64 %a)
```

```
    ret i64 %1
```

```
}
```

```
; RUN: llc < %s -mtriple=s390x-linux-gnu -mcpu=zEC12 -verify-machineinstrs | FileCheck %s
```

```
;
```

```
; Test that early if conversion produces LOCR with operands of the right
```

```
; register classes.
```

```
define void @autogen_SD4739(ptr) {
```

; CHECK-NOT: Expected a GR32Bit register, but got a GRX32Bit register

BB:

%L34 = load i8, ptr %0

%Cmp56 = icmp sgt i8 undef, %L34

br label %CF246

CF246: ; preds = %CF246, %BB

%S1163 = select i1 %Cmp56, i8 %L34, i8 undef

br i1 undef, label %CF246, label %CF248

CF248: ; preds = %CF248, %CF246

store i8 %S1163, ptr %0

br label %CF248

}

; RUN: llc -march=hexagon -hexagon-hvx-widen=32 < %s | FileCheck %s

; Truncating a type-to-be-widenened to a legal type (v8i8).

; Check that this compiles successfully.

; CHECK-LABEL: f0:

; CHECK: dealloc\_return

target datalayout = "e-m:e-p:32:32:32-a:0-n16:32-i64:64:64-i32:32:32-i16:16:16-i1:8:8-f32:32:32-f64:64:64-v32:32:32-v64:64:64-v512:512:512-v1024:1024:1024-v2048:2048:2048"

target triple = "hexagon"

define dlllexport void @f0(ptr %a0) local\_unnamed\_addr #0 {

b0:

%v0 = load i8, ptr undef, align 1

%v1 = zext i8 %v0 to i16

%v2 = add i16 0, %v1

%v3 = icmp sgt i16 %v2, 1

%v4 = select i1 %v3, i16 %v2, i16 1

%v5 = udiv i16 -32768, %v4

%v6 = zext i16 %v5 to i32

%v7 = insertelement <8 x i32> undef, i32 %v6, i32 0

%v8 = shufflevector <8 x i32> %v7, <8 x i32> undef, <8 x i32> zeroinitializer

%v9 = load <8 x i16>, ptr undef, align 2

%v10 = sext <8 x i16> %v9 to <8 x i32>

%v11 = mul nsw <8 x i32> %v8, %v10

%v12 = add nsw <8 x i32> %v11, <i32 16384, i32 16384, i32 16384, i32 16384, i32 16384, i32 16384, i32 16384, i32 16384>

%v13 = lshr <8 x i32> %v12, <i32 15, i32 15, i32 15, i32 15, i32 15, i32 15, i32 15, i32 15>

%v14 = trunc <8 x i32> %v13 to <8 x i8>

%v15 = getelementptr inbounds i8, ptr %a0, i32 undef

store <8 x i8> %v14, ptr %v15, align 1

ret void

}

```
attributes #0 = { "target-features"="+hvx,+hvx-length128b" }  
; RUN: llc -O3 -mtriple=powerpc-unknown-linux-gnu -mcpu=e500 -mattr=spe < %s | FileCheck %s
```

```
; PowerPC SPE is a rare in-tree target that has the FP_TO_SINT node marked  
; as Legal.
```

```
; Verify that fptosi(42.1) isn't simplified when the rounding mode is  
; unknown.  
; Verify that no gross errors happen.  
; CHECK-LABEL: @f20  
; COMMON: cfdctsiz  
define i32 @f20(double %a) strictfp {  
entry:  
  %result = call i32 @llvm.experimental.constrained.fptosi.i32.f64(double 42.1,  
    metadata !"fpexcept.strict")  
    strictfp  
  ret i32 %result  
}
```

```
@llvm.fp.env = thread_local global i8 zeroinitializer, section "llvm.metadata"  
declare i32 @llvm.experimental.constrained.fptosi.i32.f64(double, metadata)  
Copyright (c) 2019 Intel Corporation. All rights reserved.
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
; NOTE: Assertions have been autogenerated by utils/update_analyze_test_checks.py  
; RUN: opt < %s -passes="print<cost-model>" 2>&1 -disable-output -mtriple=x86_64-apple-macosx10.8.0 -  
mattr=+avx2 | FileCheck %s --check-prefixes=VEC256,AVX  
; RUN: opt < %s -passes="print<cost-model>" 2>&1 -disable-output -mtriple=x86_64-apple-macosx10.8.0 -  
mattr=+avx512vl,+prefer-256-bit | FileCheck %s --check-prefixes=VEC256,AVX512VL256
```

```

; RUN: opt < %s -passes="print<cost-model>" 2>&1 -disable-output -mtriple=x86_64-apple-macosx10.8.0 -
mattr=+avx512vl,-prefer-256-bit | FileCheck %s --check-prefixes=AVX512VL512
; RUN: opt < %s -passes="print<cost-model>" 2>&1 -disable-output -mtriple=x86_64-apple-macosx10.8.0 -
mattr=+avx512vl,+avx512bw,+avx512dq,+prefer-256-bit | FileCheck %s --check-prefixes=VEC256,SKX256
; RUN: opt < %s -passes="print<cost-model>" 2>&1 -disable-output -mtriple=x86_64-apple-macosx10.8.0 -
mattr=+avx512vl,+avx512bw,+avx512dq,-prefer-256-bit | FileCheck %s --check-prefixes=SKX512

define
void @zext256() "min-legal-vector-width"="256" {
; AVX-LABEL: 'zext256'
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %A = zext <8 x i16> undef to <8 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %B = zext <8 x i32> undef to <8 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %C = zext <16 x i8> undef to <16 x i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %D = zext <16 x i16> undef to <16 x
i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %E = zext <32 x i8> undef to <32 x i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL256-LABEL: 'zext256'
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %A = zext <8 x i16> undef to
<8 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %B = zext <8 x i32> undef to
<8 x i64>
; AVX512VL256-NEXT: Cost
Model: Found an estimated cost of 2 for instruction: %C = zext <16 x i8> undef to <16 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %D = zext <16 x i16> undef to
<16 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %E = zext <32 x i8> undef to
<32 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL512-LABEL: 'zext256'
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = zext <8 x i16> undef to
<8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = zext <8 x i32> undef to
<8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = zext <16 x i8> undef to
<16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = zext <16 x i16> undef to
<16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction:
%E = zext <32 x i8> undef to <32 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX256-LABEL: 'zext256'
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %A = zext <8 x i16> undef to <8 x
i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %B = zext <8 x i32> undef to <8 x

```



```

i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %C = zext <16 x i8> undef to <16 x
i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %D = zext <16 x i16> undef to <16 x
i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %E = zext <32 x i8> undef to <32 x
i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX512-LABEL: 'zext256'
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = zext <8 x i16> undef to <8 x
i64>
; SKX512-NEXT: Cost Model: Found an estimated
cost of 1 for instruction: %B = zext <8 x i32> undef to <8 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = zext <16 x i8> undef to <16 x
i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = zext <16 x i16> undef to <16 x
i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = zext <32 x i8> undef to <32 x
i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
%A = zext <8 x i16> undef to <8 x i64>
%B = zext <8 x i32> undef to <8 x i64>
%C = zext <16 x i8> undef to <16 x i32>
%D = zext <16 x i16> undef to <16 x i32>
%E = zext <32 x i8> undef to <32 x i16>
ret void
}

```

```

define void @zext512() "min-legal-vector-width"="512" {
; AVX-LABEL: 'zext512'
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %A = zext <8 x i16> undef to <8 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %B = zext <8 x i32>
undef to <8 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %C = zext <16 x i8> undef to <16 x i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %D = zext <16 x i16> undef to <16 x
i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %E = zext <32 x i8> undef to <32 x i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL256-LABEL: 'zext512'
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = zext <8 x i16> undef to
<8 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = zext <8 x i32> undef to
<8 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = zext <16 x i8> undef to
<16 x i32>

```

```

; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = zext <16 x i16> undef to
<16 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost
of 3 for instruction: %E = zext <32 x i8> undef to <32 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL512-LABEL: 'zext512'
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = zext <8 x i16> undef to
<8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = zext <8 x i32> undef to
<8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = zext <16 x i8> undef to
<16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = zext <16 x i16> undef to
<16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %E = zext <32 x i8> undef to
<32 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX256-LABEL: 'zext512'
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = zext <8 x i16> undef to <8
x i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = zext <8 x i32> undef to <8 x
i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = zext <16 x i8> undef to <16 x
i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = zext <16 x i16> undef to <16 x
i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = zext <32 x i8> undef to <32 x
i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX512-LABEL: 'zext512'
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = zext <8 x i16> undef to <8 x
i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = zext <8 x i32> undef to <8 x
i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = zext <16 x i8> undef to <16 x
i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = zext <16
x i16> undef to <16 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = zext <32 x i8> undef to <32 x
i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
%A = zext <8 x i16> undef to <8 x i64>
%B = zext <8 x i32> undef to <8 x i64>
%C = zext <16 x i8> undef to <16 x i32>

```

```

%D = zext <16 x i16> undef to <16 x i32>
%E = zext <32 x i8> undef to <32 x i16>
ret void
}

define void @sext256() "min-legal-vector-width"="256" {
; AVX-LABEL: 'sext256'
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %A = sext <8 x i8> undef to <8 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %B = sext <8 x i16> undef to <8 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %C = sext <8 x i32> undef to <8 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %D = sext <16 x i8> undef to <16 x i32>
; AVX-NEXT: Cost Model: Found
an estimated cost of 3 for instruction: %E = sext <16 x i16> undef to <16 x i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %F = sext <32 x i8> undef to <32 x i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL256-LABEL: 'sext256'
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %A = sext <8 x i8> undef to <8
x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %B = sext <8 x i16> undef to
<8 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %C = sext <8 x i32> undef to
<8 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %D = sext <16 x i8> undef to
<16 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %E = sext <16 x i16> undef to
<16 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %F = sext <32 x i8> undef
to <32 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL512-LABEL: 'sext256'
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = sext <8 x i8> undef to <8
x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to
<8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to
<8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = sext <16 x i8> undef to
<16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = sext <16 x i16> undef to
<16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %F = sext <32 x i8> undef to
<32 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX256-LABEL: 'sext256'
; SKX256-NEXT: Cost Model: Found an

```

```

estimated cost of 2 for instruction: %A = sext <8 x i8> undef to <8 x i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %B = sext <8 x i16> undef to <8 x i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %C = sext <8 x i32> undef to <8 x i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %D = sext <16 x i8> undef to <16 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %E = sext <16 x i16> undef to <16 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %F = sext <32 x i8> undef to <32 x i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX512-LABEL: 'sext256'
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = sext <8 x i8> undef to <8 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to <8 x i64>
; SKX512-NEXT: Cost
Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to <8 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = sext <16 x i8> undef to <16 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = sext <16 x i16> undef to <16 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %F = sext <32 x i8> undef to <32 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
%A = sext <8 x i8> undef to <8 x i64>
%B = sext <8 x i16> undef to <8 x i64>
%C = sext <8 x i32> undef to <8 x i64>
%D = sext <16 x i8> undef to <16 x i32>
%E = sext <16 x i16> undef to <16 x i32>
%F = sext <32 x i8> undef to <32 x i16>
ret void
}

```

```

define void @sext512() "min-legal-vector-width"="512" {
; AVX-LABEL: 'sext512'
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %A = sext <8 x i8> undef to <8 x i64>
; AVX-NEXT: Cost Model: Found
an estimated cost of 4 for instruction: %B = sext <8 x i16> undef to <8 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %C = sext <8 x i32> undef to <8 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %D = sext <16 x i8> undef to <16 x i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %E = sext <16 x i16> undef to <16 x i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %F = sext <32 x i8> undef to <32 x i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL256-LABEL: 'sext512'

```

```

; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = sext <8 x i8> undef to <8
x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to
<8 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to
<8 x i64>
; AVX512VL256-NEXT:
  Cost Model: Found an estimated cost of 1 for instruction: %D = sext <16 x i8> undef to <16 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = sext <16 x i16> undef to
<16 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %F = sext <32 x i8> undef to
<32 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL512-LABEL: 'sext512'
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = sext <8 x i8> undef to <8
x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to
<8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to
<8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = sext <16 x i8> undef to
<16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost
of 1 for instruction: %E = sext <16 x i16> undef to <16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %F = sext <32 x i8> undef to
<32 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX256-LABEL: 'sext512'
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = sext <8 x i8> undef to <8 x i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to <8 x
i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to <8 x
i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = sext <16 x i8> undef to <16 x
i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = sext <16 x i16> undef to <16 x
i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %F = sext <32 x i8> undef to <32 x
i16>
; SKX256-NEXT: Cost
  Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX512-LABEL: 'sext512'
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = sext <8 x i8> undef to <8 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to <8 x
i64>

```

```

; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to <8 x
i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = sext <16 x i8> undef to <16 x
i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = sext <16 x i16> undef to <16 x
i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %F = sext <32 x i8> undef to <32 x
i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
%A = sext <8 x i8> undef to <8 x i64>
%B = sext <8 x i16> undef to <8 x i64>
%C = sext <8 x i32> undef to <8 x i64>
%D = sext <16 x i8>
undef to <16 x i32>
%E = sext <16 x i16> undef to <16 x i32>
%F = sext <32 x i8> undef to <32 x i16>
ret void
}

define void @trunc256() "min-legal-vector-width"="256" {
; VEC256-LABEL: 'trunc256'
; VEC256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %A = trunc <8 x i64> undef to <8 x
i32>
; VEC256-NEXT: Cost Model: Found an estimated cost of 10 for instruction: %B = trunc <8 x i64> undef to <8 x
i16>
; VEC256-NEXT: Cost Model: Found an estimated cost of 8 for instruction: %C = trunc <8 x i64> undef to <8 x
i8>
; VEC256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %D = trunc <16 x i32> undef to <16 x
i16>
; VEC256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %E = trunc <16 x i32> undef to <16 x
i8>
; VEC256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %F = trunc <32 x i16> undef to <32 x
i8>
; VEC256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL512-LABEL: 'trunc256'
;
AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = trunc <8 x i64> undef to
<8 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %B = trunc <8 x i64> undef to
<8 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %C = trunc <8 x i64> undef to
<8 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %D = trunc <16 x i32> undef
to <16 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %E = trunc <16 x i32> undef to
<16 x i8>

```

```

; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 8 for instruction: %F = trunc <32 x i16> undef to
<32 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX512-LABEL: 'trunc256'
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = trunc <8 x i64> undef to <8 x
i32>
; SKX512-NEXT: Cost Model: Found an estimated
cost of 2 for instruction: %B = trunc <8 x i64> undef to <8 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %C = trunc <8 x i64> undef to <8 x
i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %D = trunc <16 x i32> undef to <16 x
i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %E = trunc <16 x i32> undef to <16 x
i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %F = trunc <32 x i16> undef to <32 x
i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
%A = trunc <8 x i64> undef to <8 x i32>
%B = trunc <8 x i64> undef to <8 x i16>
%C = trunc <8 x i64> undef to <8 x i8>
%D = trunc <16 x i32> undef to <16 x i16>
%E = trunc <16 x i32> undef to <16 x i8>
%F = trunc <32 x i16> undef to <32 x i8>
ret void
}

```

```

define i32 @zext256_vXi1() "min-legal-vector-width"="256" {
; AVX-LABEL: 'zext256_vXi1'
; AVX-NEXT: Cost Model: Found
an estimated cost of 1 for instruction: %V2i64 = zext <2 x i1> undef to <2 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V4i64 = zext <4 x i1> undef to <4 x
i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V8i64 = zext <8 x i1> undef to <8 x
i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i32 = zext <2 x i1> undef to <2 x
i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i32 = zext <4 x i1> undef to <4 x
i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V8i32 = zext <8 x i1> undef to <8 x
i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V16i32 = zext <16 x i1> undef to <16 x
i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i16 = zext <2 x i1> undef to <2 x
i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i16 = zext <4 x i1> undef to <4 x
i16>

```

```

; AVX-NEXT:
  Cost Model: Found an estimated cost of 1 for instruction: %V8i16 = zext <8 x i1> undef to <8 x i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i16 = zext <16 x i1> undef to <16 x
i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V32i16 = zext <32 x i1> undef to <32 x
i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i8 = zext <2 x i1> undef to <2 x i8>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i8 = zext <4 x i1> undef to <4 x i8>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i8 = zext <8 x i1> undef to <8 x i8>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i8 = zext <16 x i1> undef to <16 x
i8>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V32i8 = zext <32 x i1> undef to <32 x
i8>
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V64i8 = zext <64 x i1> undef
to <64 x i8>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; AVX512VL256-LABEL: 'zext256_vXi1'
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = zext <2 x i1> undef
to <2 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = zext <4 x i1> undef
to <4 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V8i64 = zext <8 x i1> undef
to <8 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i32 = zext <2 x i1> undef
to <2 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = zext <4 x i1> undef
to <4 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = zext <8 x i1> undef
to <8 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V16i32 = zext <16 x i1>
undef to <16 x i32>
; AVX512VL256-NEXT:
  Cost Model: Found an estimated cost of 5 for instruction: %V2i16 = zext <2 x i1> undef to <2 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V4i16 = zext <4 x i1> undef
to <4 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V8i16 = zext <8 x i1> undef
to <8 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 12 for instruction: %V16i16 = zext <16 x i1>
undef to <16 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 24 for instruction: %V32i16 = zext <32 x i1>
undef to <32 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 6 for instruction: %V2i8 = zext <2 x i1> undef to
<2 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 6 for instruction: %V4i8 = zext <4 x i1> undef to
<4 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 6 for instruction: %V8i8 = zext <8 x i1> undef to
<8 x i8>

```



```

; AVX512VL256-NEXT: Cost Model: Found an estimated
cost of 12 for instruction: %V16i8 = zext <16 x i1> undef to <16 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 25 for instruction: %V32i8 = zext <32 x i1>
undef to <32 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 50 for instruction: %V64i8 = zext <64 x i1>
undef to <64 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; AVX512VL512-LABEL: 'zext256_vXi1'
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = zext <2 x i1> undef
to <2 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = zext <4 x i1> undef
to <4 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i64 = zext <8 x i1> undef
to <8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i32 = zext <2 x i1> undef
to <2 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of
2 for instruction: %V4i32 = zext <4 x i1> undef to <4 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = zext <8 x i1> undef
to <8 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i32 = zext <16 x i1>
undef to <16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V2i16 = zext <2 x i1> undef
to <2 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V4i16 = zext <4 x i1> undef
to <4 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V8i16 = zext <8 x i1> undef
to <8 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V16i16 = zext <16 x i1>
undef to <16 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 9 for instruction: %V32i16 = zext <32 x i1>
undef to <32 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V2i8
= zext <2 x i1> undef to <2 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V4i8 = zext <4 x i1> undef to
<4 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V8i8 = zext <8 x i1> undef to
<8 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V16i8 = zext <16 x i1> undef
to <16 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 9 for instruction: %V32i8 = zext <32 x i1> undef
to <32 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 19 for instruction: %V64i8 = zext <64 x i1>
undef to <64 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; SKX256-LABEL: 'zext256_vXi1'

```

```

; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = zext <2 x i1> undef to <2 x
i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = zext <4 x i1> undef to <4 x
i64>
;
SKX256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V8i64 = zext <8 x i1> undef to <8 x
i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i32 = zext <2 x i1> undef to <2 x
i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = zext <4 x i1> undef to <4 x
i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = zext <8 x i1> undef to <8 x
i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V16i32 = zext <16 x i1> undef to
<16 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i16 = zext <2 x i1> undef to <2 x
i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i16 = zext <4 x i1> undef to <4 x
i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i16 = zext <8 x i1> undef to <8 x
i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction:
%V16i16 = zext <16 x i1> undef to <16 x i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V32i16 = zext <32 x i1> undef to
<32 x i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i8 = zext <2 x i1> undef to <2 x
i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i8 = zext <4 x i1> undef to <4 x
i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i8 = zext <8 x i1> undef to <8 x
i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i8 = zext <16 x i1> undef to <16
x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V32i8 = zext <32 x i1> undef to <32
x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V64i8 = zext <64 x i1> undef to <64
x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; SKX512-LABEL: 'zext256_vXi1'
; SKX512-NEXT: Cost
Model: Found an estimated cost of 2 for instruction: %V2i64 = zext <2 x i1> undef to <2 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = zext <4 x i1> undef to <4 x
i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i64 = zext <8 x i1> undef to <8 x
i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i32 = zext <2 x i1> undef to <2 x
i32>

```

```

; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = zext <4 x i1> undef to <4 x
i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = zext <8 x i1> undef to <8 x
i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i32 = zext <16 x i1> undef to
<16 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i16 = zext <2 x i1> undef to <2 x
i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i16 = zext
<4 x i1> undef to <4 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i16 = zext <8 x i1> undef to <8 x
i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i16 = zext <16 x i1> undef to
<16 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V32i16 = zext <32 x i1> undef to
<32 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i8 = zext <2 x i1> undef to <2 x
i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i8 = zext <4 x i1> undef to <4 x
i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i8 = zext <8 x i1> undef to <8 x
i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i8 = zext <16 x i1> undef to <16
x i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V32i8 = zext <32 x i1> undef to <32
x i8>
; SKX512-NEXT: Cost Model: Found an estimated
cost of 2 for instruction: %V64i8 = zext <64 x i1> undef to <64 x i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
%V2i64 = zext <2 x i1> undef to <2 x i64>
%V4i64 = zext <4 x i1> undef to <4 x i64>
%V8i64 = zext <8 x i1> undef to <8 x i64>

%V2i32 = zext <2 x i1> undef to <2 x i32>
%V4i32 = zext <4 x i1> undef to <4 x i32>
%V8i32 = zext <8 x i1> undef to <8 x i32>
%V16i32 = zext <16 x i1> undef to <16 x i32>

%V2i16 = zext <2 x i1> undef to <2 x i16>
%V4i16 = zext <4 x i1> undef to <4 x i16>
%V8i16 = zext <8 x i1> undef to <8 x i16>
%V16i16 = zext <16 x i1> undef to <16 x i16>
%V32i16 = zext <32 x i1> undef to <32 x i16>

%V2i8 = zext <2 x i1> undef to <2 x i8>
%V4i8 = zext <4 x i1> undef to <4 x i8>
%V8i8 = zext <8 x i1> undef to <8 x i8>

```

%V16i8 = zext <16 x i1> undef to <16 x i8>

%V32i8 = zext <32 x i1> undef to <32 x i8>

%V64i8 = zext <64 x i1> undef to <64 x i8>

ret i32 undef

}

define i32 @sext256\_vXi1()

"min-legal-vector-width"="256" {

; AVX-LABEL: 'sext256\_vXi1'

; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I64 = sext i1 undef to i64

; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = sext <2 x i1> undef to <2 x i64>

; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V4i64 = sext <4 x i1> undef to <4 x i64>

; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V8i64 = sext <8 x i1> undef to <8 x i64>

; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I32 = sext i1 undef to i32

; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i32 = sext <2 x i1> undef to <2 x i32>

; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = sext <4 x i1> undef to <4 x i32>

; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V8i32 = sext <8 x i1> undef to <8 x i32>

; AVX-NEXT: Cost Model: Found an estimated cost  
of 4 for instruction: %V16i32 = sext <16 x i1> undef to <16 x i32>

; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I16 = sext i1 undef to i16

; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i16 = sext <2 x i1> undef to <2 x i16>

; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i16 = sext <4 x i1> undef to <4 x i16>

; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i16 = sext <8 x i1> undef to <8 x i16>

; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i16 = sext <16 x i1> undef to <16 x i16>

; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V32i16 = sext <32 x i1> undef to <32 x i16>

; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I8 = sext i1 undef to i8

; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i8 = sext <2 x i1> undef to <2 x i8>

; AVX-NEXT: Cost Model: Found an estimated cost of 2 for  
instruction: %V4i8 = sext <4 x i1> undef to <4 x i8>

; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i8 = sext <8 x i1> undef to <8 x i8>

; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i8 = sext <16 x i1> undef to <16 x i8>

; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V32i8 = sext <32 x i1> undef to <32 x i8>

; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V64i8 = sext <64 x i1> undef to <64 x i8>

```

; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; AVX512VL256-LABEL: 'sext256_vXi1'
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I64 = sext i1 undef to i64
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i64 = sext <2 x i1> undef
to <2 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i64 = sext <4 x i1> undef
to <4 x i64>
; AVX512VL256-NEXT:
Cost Model: Found an estimated cost of 3 for instruction: %V8i64 = sext <8 x i1> undef to <8 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I32 = sext i1 undef to i32
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i32 = sext <2 x i1> undef
to <2 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i32 = sext <4 x i1> undef
to <4 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i32 = sext <8 x i1> undef
to <8 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i32 = sext <16 x i1>
undef to <16 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I16 = sext i1 undef to i16
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V2i16 = sext <2 x i1> undef
to <2 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 4 for instruction:
%V4i16 = sext <4 x i1> undef to <4 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V8i16 = sext <8 x i1> undef
to <8 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 10 for instruction: %V16i16 = sext <16 x i1>
undef to <16 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 20 for instruction: %V32i16 = sext <32 x i1>
undef to <32 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I8 = sext i1 undef to i8
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V2i8 = sext <2 x i1> undef to
<2 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V4i8 = sext <4 x i1> undef to
<4 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V8i8 = sext <8 x i1> undef to
<8 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 10 for instruction: %V16i8 = sext <16 x i1>
undef to <16 x i8>
;
AVX512VL256-NEXT: Cost Model: Found an estimated cost of 21 for instruction: %V32i8 = sext <32 x i1>
undef to <32 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 42 for instruction: %V64i8 = sext <64 x i1>
undef to <64 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; AVX512VL512-LABEL: 'sext256_vXi1'
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I64 = sext i1 undef to i64

```

```

; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i64 = sext <2 x i1> undef
to <2 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i64 = sext <4 x i1> undef
to <4 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i64 = sext <8 x i1> undef
to <8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I32 = sext i1 undef to i32
; AVX512VL512-NEXT: Cost Model: Found an estimated
cost of 1 for instruction: %V2i32 = sext <2 x i1> undef to <2 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i32 = sext <4 x i1> undef
to <4 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i32 = sext <8 x i1> undef
to <8 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i32 = sext <16 x i1>
undef to <16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I16 = sext i1 undef to i16
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V2i16 = sext <2 x i1> undef
to <2 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V4i16 = sext <4 x i1> undef
to <4 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V8i16 = sext <8 x i1> undef
to <8 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V16i16 =
sext <16 x i1> undef to <16 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 7 for instruction: %V32i16 = sext <32 x i1>
undef to <32 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I8 = sext i1 undef to i8
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V2i8 = sext <2 x i1> undef to
<2 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V4i8 = sext <4 x i1> undef to
<4 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V8i8 = sext <8 x i1> undef to
<8 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V16i8 = sext <16 x i1> undef
to <16 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 7 for instruction: %V32i8 = sext <32 x i1> undef
to <32 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 15 for instruction: %V64i8 = sext <64 x i1>
undef to <64 x i8>
; AVX512VL512-NEXT:
Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; SKX256-LABEL: 'sext256_vXi1'
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I64 = sext i1 undef to i64
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i64 = sext <2 x i1> undef to <2 x
i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i64 = sext <4 x i1> undef to <4 x
i64>

```

```

; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i64 = sext <8 x i1> undef to <8 x
i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I32 = sext i1 undef to i32
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i32 = sext <2 x i1> undef to <2 x
i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i32 = sext <4 x i1> undef to <4 x
i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i32 = sext <8 x i1> undef
to <8 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i32 = sext <16 x i1> undef to
<16 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I16 = sext i1 undef to i16
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i16 = sext <2 x i1> undef to <2 x
i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i16 = sext <4 x i1> undef to <4 x
i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i16 = sext <8 x i1> undef to <8 x
i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i16 = sext <16 x i1> undef to
<16 x i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V32i16 = sext <32 x i1> undef to
<32 x i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I8 = sext i1 undef to i8
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i8
= sext <2 x i1> undef to <2 x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i8 = sext <4 x i1> undef to <4 x
i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i8 = sext <8 x i1> undef to <8 x
i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i8 = sext <16 x i1> undef to <16
x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V32i8 = sext <32 x i1> undef to <32
x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V64i8 = sext <64 x i1> undef to <64
x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; SKX512-LABEL: 'sext256_vXi1'
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I64 = sext i1 undef to i64
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i64 = sext <2 x i1> undef to <2 x
i64>
; SKX512-NEXT: Cost Model: Found an estimated
cost of 1 for instruction: %V4i64 = sext <4 x i1> undef to <4 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i64 = sext <8 x i1> undef to <8 x
i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I32 = sext i1 undef to i32
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i32 = sext <2 x i1> undef to <2 x
i32>

```

```

; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i32 = sext <4 x i1> undef to <4 x
i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i32 = sext <8 x i1> undef to <8 x
i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i32 = sext <16 x i1> undef to
<16 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I16 = sext i1 undef to i16
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i16 = sext <2 x i1> undef to <2 x
i16>
; SKX512-NEXT: Cost Model:
Found an estimated cost of 1 for instruction: %V4i16 = sext <4 x i1> undef to <4 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i16 = sext <8 x i1> undef to <8 x
i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i16 = sext <16 x i1> undef to
<16 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V32i16 = sext <32 x i1> undef to
<32 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I8 = sext i1 undef to i8
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i8 = sext <2 x i1> undef to <2 x
i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i8 = sext <4 x i1> undef to <4 x
i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i8 = sext <8 x i1> undef to <8 x
i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i8 = sext <16 x i1> undef to <16
x i8>
;
SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V32i8 = sext <32 x i1> undef to <32
x i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V64i8 = sext <64 x i1> undef to <64
x i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
%I64 = sext i1 undef to i64
%V2i64 = sext <2 x i1> undef to <2 x i64>
%V4i64 = sext <4 x i1> undef to <4 x i64>
%V8i64 = sext <8 x i1> undef to <8 x i64>

%I32 = sext i1 undef to i32
%V2i32 = sext <2 x i1> undef to <2 x i32>
%V4i32 = sext <4 x i1> undef to <4 x i32>
%V8i32 = sext <8 x i1> undef to <8 x i32>
%V16i32 = sext <16 x i1> undef to <16 x i32>

%I16 = sext i1 undef to i16
%V2i16 = sext <2 x i1> undef to <2 x i16>
%V4i16 = sext <4 x i1> undef to <4 x i16>
%V8i16 = sext <8 x i1> undef to <8 x i16>

```



%V16i16 = sext <16 x i1> undef to <16 x i16>

%V32i16 = sext <32 x i1> undef to <32 x i16>

%I8 = sext i1 undef to i8

%V2i8 = sext <2

x i1> undef to <2 x i8>

%V4i8 = sext <4 x i1> undef to <4 x i8>

%V8i8 = sext <8 x i1> undef to <8 x i8>

%V16i8 = sext <16 x i1> undef to <16 x i8>

%V32i8 = sext <32 x i1> undef to <32 x i8>

%V64i8 = sext <64 x i1> undef to <64 x i8>

ret i32 undef

}

define i32 @trunc\_vXi1() "min-legal-vector-width"="256" {

; AVX-LABEL: 'trunc\_vXi1'

; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V2i64 = trunc <2 x i64> undef to <2 x i1>

; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V4i64 = trunc <4 x i64> undef to <4 x i1>

; AVX-NEXT: Cost Model: Found an estimated cost of 9 for instruction: %V8i64 = trunc <8 x i64> undef to <8 x i1>

; AVX-NEXT: Cost Model: Found an estimated cost of 11 for instruction: %V16i64 = trunc <16 x i64> undef to <16 x i1>

; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i64 = trunc <32 x i64> undef to <32 x i1>

; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V64i64 = trunc <64 x i64> undef to <64 x i1>

; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i32 = trunc <2 x i32> undef to <2 x i1>

; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V4i32 = trunc <4 x i32> undef to <4 x i1>

; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = trunc <8 x i32> undef to <8 x i1>

; AVX-NEXT: Cost Model: Found an estimated cost of 8 for instruction: %V16i32 = trunc <16 x i32> undef to <16 x i1>

; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i32 = trunc <32 x i32> undef to <32 x i1>

; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V64i32 = trunc <64 x i32> undef to <64 x i1>

; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i16 = trunc <2 x i16> undef to <2 x i1>

; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i16 = trunc <4 x i16> undef to <4 x i1>

; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V8i16 = trunc <8 x i16> undef to <8 x i1>

; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V16i16 = trunc <16 x i16> undef to <16

```

x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i16 = trunc <32 x i16> undef to <32
x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V64i16 = trunc <64 x i16> undef to <64
x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i8 = trunc <2 x i8> undef to <2 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i8 = trunc <4 x i8> undef to <4 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i8 = trunc <8 x i8> undef to <8 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V16i8 = trunc <16 x i8> undef to <16 x
i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i8 = trunc <32 x i8> undef to <32 x
i1>
; AVX-NEXT:
  Cost Model: Found an estimated cost of 0 for instruction: %V64i8 = trunc <64 x i8> undef to <64 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; AVX512VL256-LABEL: 'trunc_vXi1'
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = trunc <2 x i64>
undef to <2 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = trunc <4 x i64>
undef to <4 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 9 for instruction: %V8i64 = trunc <8 x i64>
undef to <8 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 11 for instruction: %V16i64 = trunc <16 x i64>
undef to <16 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i64 = trunc <32 x i64>
undef to <32 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V64i64 = trunc <64 x i64>
undef to <64 x i1>
; AVX512VL256-NEXT: Cost Model:
  Found an estimated cost of 2 for instruction: %V2i32 = trunc <2 x i32> undef to <2 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = trunc <4 x i32>
undef to <4 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = trunc <8 x i32>
undef to <8 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V16i32 = trunc <16 x i32>
undef to <16 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 8 for instruction: %V32i32 = trunc <32 x i32>
undef to <32 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 16 for instruction: %V64i32 = trunc <64 x i32>
undef to <64 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V2i16 = trunc <2 x i16>
undef to <2 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V4i16 = trunc <4 x i16>
undef to <4 x i1>
; AVX512VL256-NEXT: Cost Model: Found an
  estimated cost of 3 for instruction: %V8i16 = trunc <8 x i16> undef to <8 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 8 for instruction: %V16i16 = trunc <16 x i16>

```

```

undef to <16 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 16 for instruction: %V32i16 = trunc <32 x i16>
undef to <32 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 32 for instruction: %V64i16 = trunc <64 x i16>
undef to <64 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V2i8 = trunc <2 x i8> undef
to <2 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V4i8 = trunc <4 x i8> undef
to <4 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V8i8 = trunc <8 x i8> undef
to <8 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 8 for instruction: %V16i8 = trunc <16 x i8>
undef to <16 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of
0 for instruction: %V32i8 = trunc <32 x i8> undef to <32 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V64i8 = trunc <64 x i8>
undef to <64 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; AVX512VL512-LABEL: 'trunc_vXi1'
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = trunc <2 x i64>
undef to <2 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = trunc <4 x i64>
undef to <4 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i64 = trunc <8 x i64>
undef to <8 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 11 for instruction: %V16i64 = trunc <16 x i64>
undef to <16 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i64 = trunc <32 x i64>
undef to <32 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction:
%V64i64 = trunc <64 x i64> undef to <64 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i32 = trunc <2 x i32>
undef to <2 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = trunc <4 x i32>
undef to <4 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = trunc <8 x i32>
undef to <8 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i32 = trunc <16 x i32>
undef to <16 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V32i32 = trunc <32 x i32>
undef to <32 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 8 for instruction: %V64i32 = trunc <64 x i32>
undef to <64 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V2i16 = trunc <2 x i16>
undef to <2 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction:

```

```

%V4i16 = trunc <4 x i16> undef to <4 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V8i16 = trunc <8 x i16>
undef to <8 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V16i16 = trunc <16 x i16>
undef to <16 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i16 = trunc <32 x i16>
undef to <32 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V64i16 = trunc <64 x i16>
undef to <64 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V2i8 = trunc <2 x i8> undef
to <2 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V4i8 = trunc <4 x i8> undef
to <4 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V8i8 = trunc <8 x i8> undef
to <8 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V16i8 = trunc <16 x
i8> undef to <16 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i8 = trunc <32 x i8>
undef to <32 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V64i8 = trunc <64 x i8>
undef to <64 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; SKX256-LABEL: 'trunc_vXi1'
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = trunc <2 x i64> undef to <2
x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = trunc <4 x i64> undef to <4
x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 9 for instruction: %V8i64 = trunc <8 x i64> undef to <8
x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 11 for instruction: %V16i64 = trunc <16 x i64> undef to
<16 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i64 = trunc <32 x i64> undef to
<32 x i1>
; SKX256-NEXT:
Cost Model: Found an estimated cost of 0 for instruction: %V64i64 = trunc <64 x i64> undef to <64 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i32 = trunc <2 x i32> undef to <2
x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = trunc <4 x i32> undef to <4
x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = trunc <8 x i32> undef to <8
x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V16i32 = trunc <16 x i32> undef to
<16 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i32 = trunc <32 x i32> undef to
<32 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V64i32 = trunc <64 x i32> undef to
<64 x i1>

```

```

; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i16 = trunc <2 x i16> undef to <2
x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction:
%V4i16 = trunc <4 x i16> undef to <4 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i16 = trunc <8 x i16> undef to <8
x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i16 = trunc <16 x i16> undef to
<16 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V32i16 = trunc <32 x i16> undef to
<32 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 8 for instruction: %V64i16 = trunc <64 x i16> undef to
<64 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i8 = trunc <2 x i8> undef to <2 x
i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i8 = trunc <4 x i8> undef to <4 x
i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i8 = trunc <8 x i8> undef to <8 x
i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i8 = trunc <16 x i8> undef to
<16 x i1>
; SKX256-NEXT:
Cost Model: Found an estimated cost of 2 for instruction: %V32i8 = trunc <32 x i8> undef to <32 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V64i8 = trunc <64 x i8> undef to
<64 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; SKX512-LABEL: 'trunc_vXi1'
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = trunc <2 x i64> undef to <2
x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = trunc <4 x i64> undef to <4
x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i64 = trunc <8 x i64> undef to <8
x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 11 for instruction: %V16i64 = trunc <16 x i64> undef to
<16 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i64 = trunc <32 x i64> undef to
<32 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction:
%V64i64 = trunc <64 x i64> undef to <64 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i32 = trunc <2 x i32> undef to <2
x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = trunc <4 x i32> undef to <4
x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = trunc <8 x i32> undef to <8
x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i32 = trunc <16 x i32> undef to
<16 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i32 = trunc <32 x i32> undef to

```

```

<32 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V64i32 = trunc <64 x i32> undef to
<64 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i16 = trunc <2 x i16> undef to <2
x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i16 = trunc <4 x i16> undef to <4
x i1>
;
SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i16 = trunc <8 x i16> undef to <8
x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i16 = trunc <16 x i16> undef to
<16 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V32i16 = trunc <32 x i16> undef to
<32 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V64i16 = trunc <64 x i16> undef to
<64 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i8 = trunc <2 x i8> undef to <2 x
i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i8 = trunc <4 x i8> undef to <4 x
i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i8 = trunc <8 x i8> undef to <8 x
i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i8 = trunc <16 x i8> undef to
<16 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for
instruction: %V32i8 = trunc <32 x i8> undef to <32 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V64i8 = trunc <64 x i8> undef to
<64 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
%V2i64 = trunc <2 x i64> undef to <2 x i1>
%V4i64 = trunc <4 x i64> undef to <4 x i1>
%V8i64 = trunc <8 x i64> undef to <8 x i1>
%V16i64 = trunc <16 x i64> undef to <16 x i1>
%V32i64 = trunc <32 x i64> undef to <32 x i1>
%V64i64 = trunc <64 x i64> undef to <64 x i1>

%V2i32 = trunc <2 x i32> undef to <2 x i1>
%V4i32 = trunc <4 x i32> undef to <4 x i1>
%V8i32 = trunc <8 x i32> undef to <8 x i1>
%V16i32 = trunc <16 x i32> undef to <16 x i1>
%V32i32 = trunc <32 x i32> undef to <32 x i1>
%V64i32 = trunc <64 x i32> undef to <64 x i1>

%V2i16 = trunc <2 x i16> undef to <2 x i1>
%V4i16 = trunc <4 x i16> undef to <4 x i1>
%V8i16 = trunc <8 x i16> undef to <8 x i1>
%V16i16 = trunc <16 x i16> undef to

```

```

<16 x i1>
%V32i16 = trunc <32 x i16> undef to <32 x i1>
%V64i16 = trunc <64 x i16> undef to <64 x i1>

%V2i8 = trunc <2 x i8> undef to <2 x i1>
%V4i8 = trunc <4 x i8> undef to <4 x i1>
%V8i8 = trunc <8 x i8> undef to <8 x i1>
%V16i8 = trunc <16 x i8> undef to <16 x i1>
%V32i8 = trunc <32 x i8> undef to <32 x i1>
%V64i8 = trunc <64 x i8> undef to <64 x i1>

ret i32 undef
}

```

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
 Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable



(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional

attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License

does not grant permission to use the trade

names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

### ---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section

3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

=====

Software from third parties included in the LLVM Project:

=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or `LICENSE` file at the top containing the specific license and restrictions which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every file.

=====

Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):

=====

The libc++abi library is dual licensed under both the University of Illinois "BSD-Like" license and the MIT license. As a user of this code you may choose to use it under either license. As a contributor, you agree to allow your code to be used under both.

Full text of the relevant licenses is included below.

=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2009-2019 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

<http://llvm.org>

Permission

is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- \* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====

Copyright (c) 2009-2014 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

; NOTE: Assertions have been autogenerated by utils/update\_llc\_test\_checks.py

; RUN: llc -mtriple=riscv64 -mattr=+m -verify-machineinstrs < %s \

; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64

;

; Get the actual value of the overflow bit.

;

define zeroext i1 @saddo1.i32(i32 signext %v1, i32 signext %v2, ptr %res) {

; RV64-LABEL: saddo1.i32:

; RV64: # %bb.0: # %entry

; RV64-NEXT: addw a3, a0, a1

; RV64-NEXT: slt a0, a3, a0

; RV64-NEXT: slti a1, a1, 0

; RV64-NEXT: xor a0, a1, a0

; RV64-NEXT: sw a3, 0(a2)

; RV64-NEXT: ret

entry:

%t = call {i32, i1} @llvm.sadd.with.overflow.i32(i32 %v1, i32 %v2)

%val = extractvalue {i32, i1} %t, 0

%obit = extractvalue {i32, i1} %t, 1

store i32 %val, ptr %res

ret i1 %obit

}

; Test the immediate version.

define zeroext i1 @saddo2.i32(i32 signext %v1, ptr %res) {

; RV64-LABEL: saddo2.i32:

; RV64: # %bb.0: # %entry

; RV64-NEXT: addiw a2, a0, 4

; RV64-NEXT: slt a0, a2, a0

; RV64-NEXT:

sw a2, 0(a1)

; RV64-NEXT: ret

entry:

%t = call {i32, i1} @llvm.sadd.with.overflow.i32(i32 %v1, i32 4)

%val = extractvalue {i32, i1} %t, 0

%obit = extractvalue {i32, i1} %t, 1

store i32 %val, ptr %res

ret i1 %obit

}

; Test negative immediates.

define zeroext i1 @saddo3.i32(i32 signext %v1, ptr %res) {

; RV64-LABEL: saddo3.i32:

; RV64: # %bb.0: # %entry

; RV64-NEXT: addiw a2, a0, -4

```

; RV64-NEXT:  slt a0, a2, a0
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  sw a2, 0(a1)
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.sadd.with.overflow.i32(i32 %v1, i32 -4)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

; Test immediates that are too large to be encoded.
define zeroext i1 @saddo4.i32(i32 signext %v1, ptr %res) {
; RV64-LABEL: saddo4.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  lui a2, 4096
; RV64-NEXT:  addi a2, a2, -1
; RV64-NEXT:  addw a2, a0, a2
; RV64-NEXT:  slt
a0, a2, a0
; RV64-NEXT:  sw a2, 0(a1)
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.sadd.with.overflow.i32(i32 %v1, i32 16777215)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @saddo1.i64(i64 %v1, i64 %v2, ptr %res) {
; RV64-LABEL: saddo1.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  add a3, a0, a1
; RV64-NEXT:  slt a0, a3, a0
; RV64-NEXT:  slti a1, a1, 0
; RV64-NEXT:  xor a0, a1, a0
; RV64-NEXT:  sd a3, 0(a2)
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.sadd.with.overflow.i64(i64 %v1, i64 %v2)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
store i64 %val, ptr %res
ret i1 %obit
}

```

```

define zeroext i1 @saddo2.i64(i64 %v1, ptr %res) {
; RV64-LABEL: saddo2.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addi a2, a0, 4
; RV64-NEXT:  slt a0, a2, a0
; RV64-NEXT:  sd a2, 0(a1)
; RV64-NEXT:  ret
entry:
%t = call {i64, i1}
@llvm.sadd.with.overflow.i64(i64 %v1, i64 4)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
store i64 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @saddo3.i64(i64 %v1, ptr %res) {
; RV64-LABEL: saddo3.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addi a2, a0, -4
; RV64-NEXT:  slt a0, a2, a0
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  sd a2, 0(a1)
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.sadd.with.overflow.i64(i64 %v1, i64 -4)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
store i64 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @uaddo.i32(i32 signext %v1, i32 signext %v2, ptr %res) {
; RV64-LABEL: uaddo.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addw a1, a0, a1
; RV64-NEXT:  sltu a0, a1, a0
; RV64-NEXT:  sw a1, 0(a2)
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.uadd.with.overflow.i32(i32 %v1, i32 %v2)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32,
i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

```



```

define zeroext i1 @uaddo.i32.constant(i32 signext %v1, ptr %res) {
; RV64-LABEL: uaddo.i32.constant:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addiw a2, a0, -2
; RV64-NEXT:  sltu a0, a2, a0
; RV64-NEXT:  sw a2, 0(a1)
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.uadd.with.overflow.i32(i32 %v1, i32 -2)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @uaddo.i32.constant_one(i32 signext %v1, ptr %res) {
; RV64-LABEL: uaddo.i32.constant_one:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addiw a2, a0, 1
; RV64-NEXT:  seqz a0, a2
; RV64-NEXT:  sw a2, 0(a1)
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.uadd.with.overflow.i32(i32 %v1, i32 1)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @uaddo.i64(i64 %v1, i64 %v2, ptr
%res) {
; RV64-LABEL: uaddo.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  add a1, a0, a1
; RV64-NEXT:  sltu a0, a1, a0
; RV64-NEXT:  sd a1, 0(a2)
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.uadd.with.overflow.i64(i64 %v1, i64 %v2)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
store i64 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @uaddo.i64.constant_one(i64 %v1, ptr %res) {
; RV64-LABEL: uaddo.i64.constant_one:

```

```

; RV64:    # %bb.0: # %entry
; RV64-NEXT:  addi a2, a0, 1
; RV64-NEXT:  seqz a0, a2
; RV64-NEXT:  sd a2, 0(a1)
; RV64-NEXT:  ret

entry:
%t = call {i64, i1} @llvm.uadd.with.overflow.i64(i64 %v1, i64 1)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
store i64 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @ssubo1.i32(i32 signext %v1, i32 signext %v2, ptr %res) {
; RV64-LABEL: ssubo1.i32:
; RV64:    # %bb.0: # %entry
; RV64-NEXT:  sgtz a3, a1
; RV64-NEXT:  subw a1,
a0, a1
; RV64-NEXT:  slt a0, a1, a0
; RV64-NEXT:  xor a0, a3, a0
; RV64-NEXT:  sw a1, 0(a2)
; RV64-NEXT:  ret

entry:
%t = call {i32, i1} @llvm.ssub.with.overflow.i32(i32 %v1, i32 %v2)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @ssubo2.i32(i32 signext %v1, ptr %res) {
; RV64-LABEL: ssubo2.i32:
; RV64:    # %bb.0: # %entry
; RV64-NEXT:  addiw a2, a0, 4
; RV64-NEXT:  slt a0, a2, a0
; RV64-NEXT:  sw a2, 0(a1)
; RV64-NEXT:  ret

entry:
%t = call {i32, i1} @llvm.ssub.with.overflow.i32(i32 %v1, i32 -4)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @ssubo.i64(i64 %v1, i64 %v2, ptr %res) {
; RV64-LABEL: ssubo.i64:

```

```

; RV64:    # %bb.0: # %entry
; RV64-NEXT:  sgtz a3, a1
; RV64-NEXT:  sub a1, a0, a1
; RV64-NEXT:  slt a0, a1, a0
; RV64-NEXT:  xor a0, a3, a0
; RV64-NEXT:  sd a1,
0(a2)
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.ssub.with.overflow.i64(i64 %v1, i64 %v2)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
store i64 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @usubo.i32(i32 signext %v1, i32 signext %v2, ptr %res) {
; RV64-LABEL: usubo.i32:
; RV64:    # %bb.0: # %entry
; RV64-NEXT:  subw a1, a0, a1
; RV64-NEXT:  sltu a0, a0, a1
; RV64-NEXT:  sw a1, 0(a2)
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.usub.with.overflow.i32(i32 %v1, i32 %v2)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @usubo.i32.constant.rhs(i32 signext %v1, ptr %res) {
; RV64-LABEL: usubo.i32.constant.rhs:
; RV64:    # %bb.0: # %entry
; RV64-NEXT:  addiw a2, a0, 2
; RV64-NEXT:  sltu a0, a0, a2
; RV64-NEXT:  sw a2, 0(a1)
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.usub.with.overflow.i32(i32 %v1, i32 -2)

%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @usubo.i32.constant.lhs(i32 signext %v1, ptr %res) {

```

```

; RV64-LABEL: usubo.i32.constant.lhs:
; RV64:      # %bb.0: # %entry
; RV64-NEXT: li a2, -2
; RV64-NEXT: subw a2, a2, a0
; RV64-NEXT: addi a0, a2, 1
; RV64-NEXT: seqz a0, a0
; RV64-NEXT: sw a2, 0(a1)
; RV64-NEXT: ret

entry:
%t = call {i32, i1} @llvm.usub.with.overflow.i32(i32 -2, i32 %v1)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @usubo.i64(i64 %v1, i64 %v2, ptr %res) {
; RV64-LABEL: usubo.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT: sub a1, a0, a1
; RV64-NEXT: sltu a0, a0, a1
; RV64-NEXT: sd a1, 0(a2)
; RV64-NEXT: ret

entry:
%t = call {i64, i1} @llvm.usub.with.overflow.i64(i64 %v1, i64 %v2)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t,
1
store i64 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @smulo.i32(i32 signext %v1, i32 signext %v2, ptr %res) {
; RV64-LABEL: smulo.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT: mul a1, a0, a1
; RV64-NEXT: srai a0, a1, 32
; RV64-NEXT: sraiw a3, a1, 31
; RV64-NEXT: xor a0, a0, a3
; RV64-NEXT: snez a0, a0
; RV64-NEXT: sw a1, 0(a2)
; RV64-NEXT: ret

entry:
%t = call {i32, i1} @llvm.smul.with.overflow.i32(i32 %v1, i32 %v2)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit

```

```
}
```

```
define zeroext i1 @smulo2.i32(i32 signext %v1, ptr %res) {  
; RV64-LABEL: smulo2.i32:  
; RV64:      # %bb.0: # %entry  
; RV64-NEXT:  li a2, 13  
; RV64-NEXT:  mul a2, a0, a2  
; RV64-NEXT:  srai a0, a2, 32  
; RV64-NEXT:  sraiw a3, a2, 31  
; RV64-NEXT:  xor a0, a0, a3  
; RV64-NEXT:  snez a0, a0  
; RV64-NEXT:  sw a2, 0(a1)  
; RV64-NEXT:  ret  
entry:  
%t = call {i32, i1} @llvm.smul.with.overflow.i32(i32 %v1, i32  
13)  
%val = extractvalue {i32, i1} %t, 0  
%obit = extractvalue {i32, i1} %t, 1  
store i32 %val, ptr %res  
ret i1 %obit  
}
```

```
define zeroext i1 @smulo.i64(i64 %v1, i64 %v2, ptr %res) {  
; RV64-LABEL: smulo.i64:  
; RV64:      # %bb.0: # %entry  
; RV64-NEXT:  mulh a3, a0, a1  
; RV64-NEXT:  mul a1, a0, a1  
; RV64-NEXT:  srai a0, a1, 63  
; RV64-NEXT:  xor a0, a3, a0  
; RV64-NEXT:  snez a0, a0  
; RV64-NEXT:  sd a1, 0(a2)  
; RV64-NEXT:  ret  
entry:  
%t = call {i64, i1} @llvm.smul.with.overflow.i64(i64 %v1, i64 %v2)  
%val = extractvalue {i64, i1} %t, 0  
%obit = extractvalue {i64, i1} %t, 1  
store i64 %val, ptr %res  
ret i1 %obit  
}
```

```
define zeroext i1 @smulo2.i64(i64 %v1, ptr %res) {  
; RV64-LABEL: smulo2.i64:  
; RV64:      # %bb.0: # %entry  
; RV64-NEXT:  li a2, 13  
; RV64-NEXT:  mulh a3, a0, a2  
; RV64-NEXT:  mul a2, a0, a2  
; RV64-NEXT:  srai a0, a2, 63  
; RV64-NEXT:  xor a0, a3, a0
```

```

; RV64-NEXT: snez a0, a0
; RV64-NEXT: sd a2, 0(a1)
; RV64-NEXT: ret
entry:
%t = call
{i64, i1} @llvm.smul.with.overflow.i64(i64 %v1, i64 13)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
store i64 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @umulo.i32(i32 signext %v1, i32 signext %v2, ptr %res) {
; RV64-LABEL: umulo.i32:
; RV64:    # %bb.0: # %entry
; RV64-NEXT: slli a1, a1, 32
; RV64-NEXT: slli a0, a0, 32
; RV64-NEXT: mulhu a1, a0, a1
; RV64-NEXT: srai a0, a1, 32
; RV64-NEXT: snez a0, a0
; RV64-NEXT: sw a1, 0(a2)
; RV64-NEXT: ret
entry:
%t = call {i32, i1} @llvm.umul.with.overflow.i32(i32 %v1, i32 %v2)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @umulo2.i32(i32 signext %v1, ptr %res) {
; RV64-LABEL: umulo2.i32:
; RV64:    # %bb.0: # %entry
; RV64-NEXT: li a2, 13
; RV64-NEXT: slli a2, a2, 32
; RV64-NEXT: slli a0, a0, 32
; RV64-NEXT: mulhu a2, a0, a2
; RV64-NEXT: srli a0, a2, 32
; RV64-NEXT:
    snez a0, a0
; RV64-NEXT: sw a2, 0(a1)
; RV64-NEXT: ret
entry:
%t = call {i32, i1} @llvm.umul.with.overflow.i32(i32 %v1, i32 13)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

```

```
}
```

; Similar to umulo.i32, but storing the overflow and returning the result.

```
define signext i32 @umulo3.i32(i32 signext %0, i32 signext %1, ptr %2) {  
; RV64-LABEL: umulo3.i32:  
; RV64:      # %bb.0:  
; RV64-NEXT: slli a1, a1, 32  
; RV64-NEXT: slli a0, a0, 32  
; RV64-NEXT: mulhu a0, a0, a1  
; RV64-NEXT: srai a1, a0, 32  
; RV64-NEXT: snez a1, a1  
; RV64-NEXT: sext.w a0, a0  
; RV64-NEXT: sw a1, 0(a2)  
; RV64-NEXT: ret  
%4 = tail call { i32, i1 } @llvm.umul.with.overflow.i32(i32 %0, i32 %1)  
%5 = extractvalue { i32, i1 } %4, 1  
%6 = extractvalue { i32, i1 } %4, 0  
%7 = zext i1 %5 to i32  
store i32 %7, ptr %2, align 4  
ret i32 %6  
}
```

```
define zeroext i1 @umulo.i64(i64 %v1, i64 %v2, ptr %res) {  
; RV64-LABEL: umulo.i64:  
;  
; RV64:      # %bb.0: # %entry  
; RV64-NEXT: mulhu a3, a0, a1  
; RV64-NEXT: snez a3, a3  
; RV64-NEXT: mul a0, a0, a1  
; RV64-NEXT: sd a0, 0(a2)  
; RV64-NEXT: mv a0, a3  
; RV64-NEXT: ret  
entry:  
%t = call { i64, i1 } @llvm.umul.with.overflow.i64(i64 %v1, i64 %v2)  
%val = extractvalue { i64, i1 } %t, 0  
%obit = extractvalue { i64, i1 } %t, 1  
store i64 %val, ptr %res  
ret i1 %obit  
}
```

```
define zeroext i1 @umulo2.i64(i64 %v1, ptr %res) {  
; RV64-LABEL: umulo2.i64:  
; RV64:      # %bb.0: # %entry  
; RV64-NEXT: li a3, 13  
; RV64-NEXT: mulhu a2, a0, a3  
; RV64-NEXT: snez a2, a2  
; RV64-NEXT: mul a0, a0, a3  
; RV64-NEXT: sd a0, 0(a1)
```

```

; RV64-NEXT:  mv a0, a2
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.umul.with.overflow.i64(i64 %v1, i64 13)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
store i64 %val, ptr %res
ret i1 %obit
}

;
; Check the use of the overflow bit in combination with a select instruction.
;
define i32 @saddo.select.i32(i32
  signext %v1, i32 signext %v2) {
; RV64-LABEL: saddo.select.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addw a2, a0, a1
; RV64-NEXT:  slt a2, a2, a0
; RV64-NEXT:  slti a3, a1, 0
; RV64-NEXT:  bne a3, a2, .LBB28_2
; RV64-NEXT: # %bb.1: # %entry
; RV64-NEXT:  mv a0, a1
; RV64-NEXT: .LBB28_2: # %entry
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.sadd.with.overflow.i32(i32 %v1, i32 %v2)
%obit = extractvalue {i32, i1} %t, 1
%ret = select i1 %obit, i32 %v1, i32 %v2
ret i32 %ret
}

define i1 @saddo.not.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL: saddo.not.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addw a2, a0, a1
; RV64-NEXT:  slt a0, a2, a0
; RV64-NEXT:  slti a1, a1, 0
; RV64-NEXT:  xor a0, a1, a0
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.sadd.with.overflow.i32(i32 %v1, i32 %v2)
%obit = extractvalue {i32, i1} %t, 1
%ret = xor i1 %obit, true
ret i1 %ret
}

```



```

define i64 @saddo.select.i64(i64
  %v1, i64 %v2) {
; RV64-LABEL: saddo.select.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  add a2, a0, a1
; RV64-NEXT:  slt a2, a2, a0
; RV64-NEXT:  slti a3, a1, 0
; RV64-NEXT:  bne a3, a2, .LBB30_2
; RV64-NEXT:  # %bb.1: # %entry
; RV64-NEXT:  mv a0, a1
; RV64-NEXT:  .LBB30_2: # %entry
; RV64-NEXT:  ret
entry:
  %t = call {i64, i1} @llvm.sadd.with.overflow.i64(i64 %v1, i64 %v2)
  %obit = extractvalue {i64, i1} %t, 1
  %ret = select i1 %obit, i64 %v1, i64 %v2
  ret i64 %ret
}

```

```

define i1 @saddo.not.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: saddo.not.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  add a2, a0, a1
; RV64-NEXT:  slt a0, a2, a0
; RV64-NEXT:  slti a1, a1, 0
; RV64-NEXT:  xor a0, a1, a0
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  ret
entry:
  %t = call {i64, i1} @llvm.sadd.with.overflow.i64(i64 %v1, i64 %v2)
  %obit = extractvalue {i64, i1} %t, 1
  %ret = xor i1 %obit, true
  ret i1 %ret
}

```

```

define i32 @uaddo.select.i32(i32 signext %v1,
  i32 signext %v2) {
; RV64-LABEL: uaddo.select.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addw a2, a0, a1
; RV64-NEXT:  bltu a2, a0, .LBB32_2
; RV64-NEXT:  # %bb.1: # %entry
; RV64-NEXT:  mv a0, a1
; RV64-NEXT:  .LBB32_2: # %entry
; RV64-NEXT:  ret
entry:
  %t = call {i32, i1} @llvm.uadd.with.overflow.i32(i32 %v1, i32 %v2)

```

```

%obit = extractvalue {i32, i1} %t, 1
%ret = select i1 %obit, i32 %v1, i32 %v2
ret i32 %ret
}

define i1 @uaddo.not.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL: uaddo.not.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addw a1, a0, a1
; RV64-NEXT:  sltu a0, a1, a0
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.uadd.with.overflow.i32(i32 %v1, i32 %v2)
%obit = extractvalue {i32, i1} %t, 1
%ret = xor i1 %obit, true
ret i1 %ret
}

define i64 @uaddo.select.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: uaddo.select.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  add a2, a0, a1
; RV64-NEXT:
; RV64-NEXT:  bltu a2, a0, .LBB34_2
; RV64-NEXT:  # %bb.1: # %entry
; RV64-NEXT:  mv a0, a1
; RV64-NEXT:  .LBB34_2: # %entry
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.uadd.with.overflow.i64(i64 %v1, i64 %v2)
%obit = extractvalue {i64, i1} %t, 1
%ret = select i1 %obit, i64 %v1, i64 %v2
ret i64 %ret
}

define i1 @uaddo.not.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: uaddo.not.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  add a1, a0, a1
; RV64-NEXT:  sltu a0, a1, a0
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.uadd.with.overflow.i64(i64 %v1, i64 %v2)
%obit = extractvalue {i64, i1} %t, 1
%ret = xor i1 %obit, true
ret i1 %ret
}

```

```
}
```

```
define i32 @ssubo.select.i32(i32 signext %v1, i32 signext %v2) {  
; RV64-LABEL: ssubo.select.i32:  
; RV64:      # %bb.0: # %entry  
; RV64-NEXT:  sgtz a2, a1  
; RV64-NEXT:  subw a3, a0, a1  
; RV64-NEXT:  slt a3, a3, a0  
; RV64-NEXT:  bne a2, a3, .LBB36_2  
; RV64-NEXT:  # %bb.1: # %entry  
; RV64-NEXT:  mv a0,  
a1  
; RV64-NEXT:  .LBB36_2: # %entry  
; RV64-NEXT:  ret  
entry:  
%t = call {i32, i1} @llvm.ssub.with.overflow.i32(i32 %v1, i32 %v2)  
%obit = extractvalue {i32, i1} %t, 1  
%ret = select i1 %obit, i32 %v1, i32 %v2  
ret i32 %ret  
}
```

```
define i1 @ssubo.not.i32(i32 signext %v1, i32 signext %v2) {  
; RV64-LABEL: ssubo.not.i32:  
; RV64:      # %bb.0: # %entry  
; RV64-NEXT:  sgtz a2, a1  
; RV64-NEXT:  subw a1, a0, a1  
; RV64-NEXT:  slt a0, a1, a0  
; RV64-NEXT:  xor a0, a2, a0  
; RV64-NEXT:  xori a0, a0, 1  
; RV64-NEXT:  ret  
entry:  
%t = call {i32, i1} @llvm.ssub.with.overflow.i32(i32 %v1, i32 %v2)  
%obit = extractvalue {i32, i1} %t, 1  
%ret = xor i1 %obit, true  
ret i1 %ret  
}
```

```
define i64 @ssubo.select.i64(i64 %v1, i64 %v2) {  
; RV64-LABEL: ssubo.select.i64:  
; RV64:      # %bb.0: # %entry  
; RV64-NEXT:  sgtz a2, a1  
; RV64-NEXT:  sub a3, a0, a1  
; RV64-NEXT:  slt a3, a3, a0  
; RV64-NEXT:  bne a2, a3, .LBB38_2  
; RV64-NEXT:  # %bb.1: # %entry  
; RV64-NEXT:  mv a0, a1  
; RV64-NEXT:
```

```

.LBB38_2: # %entry
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.ssub.with.overflow.i64(i64 %v1, i64 %v2)
%obit = extractvalue {i64, i1} %t, 1
%ret = select i1 %obit, i64 %v1, i64 %v2
ret i64 %ret
}

```

```

define i1 @ssub.not.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: ssub.not.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  sgtz a2, a1
; RV64-NEXT:  sub a1, a0, a1
; RV64-NEXT:  slt a0, a1, a0
; RV64-NEXT:  xor a0, a2, a0
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.ssub.with.overflow.i64(i64 %v1, i64 %v2)
%obit = extractvalue {i64, i1} %t, 1
%ret = xor i1 %obit, true
ret i1 %ret
}

```

```

define i32 @usubo.select.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL: usubo.select.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  subw a2, a0, a1
; RV64-NEXT:  bltu a0, a2, .LBB40_2
; RV64-NEXT:  # %bb.1: # %entry
; RV64-NEXT:  mv a0, a1
; RV64-NEXT:  .LBB40_2: # %entry
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.usub.with.overflow.i32(i32
%v1, i32 %v2)
%obit = extractvalue {i32, i1} %t, 1
%ret = select i1 %obit, i32 %v1, i32 %v2
ret i32 %ret
}

```

```

define i1 @usubo.not.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL: usubo.not.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  subw a1, a0, a1
; RV64-NEXT:  sltu a0, a0, a1
; RV64-NEXT:  xori a0, a0, 1

```

```

; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.usub.with.overflow.i32(i32 %v1, i32 %v2)
%obit = extractvalue {i32, i1} %t, 1
%ret = xor i1 %obit, true
ret i1 %ret
}

```

```

define i64 @usubo.select.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: usubo.select.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  sub a2, a0, a1
; RV64-NEXT:  bltu a0, a2, .LBB42_2
; RV64-NEXT:  # %bb.1: # %entry
; RV64-NEXT:  mv a0, a1
; RV64-NEXT:  .LBB42_2: # %entry
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.usub.with.overflow.i64(i64 %v1, i64 %v2)
%obit = extractvalue {i64, i1} %t, 1
%ret = select i1 %obit, i64 %v1, i64 %v2
ret i64 %ret
}

```

```

define
i1 @usubo.not.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: usubo.not.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  sub a1, a0, a1
; RV64-NEXT:  sltu a0, a0, a1
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.usub.with.overflow.i64(i64 %v1, i64 %v2)
%obit = extractvalue {i64, i1} %t, 1
%ret = xor i1 %obit, true
ret i1 %ret
}

```

```

define i32 @smulo.select.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL: smulo.select.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  mul a2, a0, a1
; RV64-NEXT:  srai a3, a2, 32
; RV64-NEXT:  sraiw a2, a2, 31
; RV64-NEXT:  bne a3, a2, .LBB44_2
; RV64-NEXT:  # %bb.1: # %entry
; RV64-NEXT:  mv a0, a1

```

```

; RV64-NEXT: .LBB44_2: # %entry
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.smul.with.overflow.i32(i32 %v1, i32 %v2)
%obit = extractvalue {i32, i1} %t, 1
%ret = select i1 %obit, i32 %v1, i32 %v2
ret i32 %ret
}

```

```

define i1 @smulo.not.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL:
smulo.not.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  mul a0, a0, a1
; RV64-NEXT:  srai a1, a0, 32
; RV64-NEXT:  sraiw a0, a0, 31
; RV64-NEXT:  xor a0, a1, a0
; RV64-NEXT:  snez a0, a0
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.smul.with.overflow.i32(i32 %v1, i32 %v2)
%obit = extractvalue {i32, i1} %t, 1
%ret = xor i1 %obit, true
ret i1 %ret
}

```

```

define i64 @smulo.select.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: smulo.select.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  mulh a2, a0, a1
; RV64-NEXT:  mul a3, a0, a1
; RV64-NEXT:  srai a3, a3, 63
; RV64-NEXT:  bne a2, a3, .LBB46_2
; RV64-NEXT: # %bb.1: # %entry
; RV64-NEXT:  mv a0, a1
; RV64-NEXT: .LBB46_2: # %entry
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.smul.with.overflow.i64(i64 %v1, i64 %v2)
%obit = extractvalue {i64, i1} %t, 1
%ret = select i1 %obit, i64 %v1, i64 %v2
ret i64 %ret
}

```

```

define i1 @smulo.not.i64(i64 %v1, i64 %v2) {
; RV64-LABEL:

```

```

smulo.not.i64:
; RV64:    # %bb.0: # %entry
; RV64-NEXT:  mulh a2, a0, a1
; RV64-NEXT:  mul a0, a0, a1
; RV64-NEXT:  srai a0, a0, 63
; RV64-NEXT:  xor a0, a2, a0
; RV64-NEXT:  snez a0, a0
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.smul.with.overflow.i64(i64 %v1, i64 %v2)
%obit = extractvalue {i64, i1} %t, 1
%ret = xor i1 %obit, true
ret i1 %ret
}

```

```

define i32 @umulo.select.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL: umulo.select.i32:
; RV64:    # %bb.0: # %entry
; RV64-NEXT:  slli a2, a1, 32
; RV64-NEXT:  slli a3, a0, 32
; RV64-NEXT:  mulhu a2, a3, a2
; RV64-NEXT:  srai a2, a2, 32
; RV64-NEXT:  bnez a2, .LBB48_2
; RV64-NEXT: # %bb.1: # %entry
; RV64-NEXT:  mv a0, a1
; RV64-NEXT: .LBB48_2: # %entry
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.umul.with.overflow.i32(i32 %v1, i32 %v2)
%obit = extractvalue {i32, i1} %t, 1
%ret = select i1 %obit, i32 %v1, i32 %v2
ret i32 %ret
}

```

```

define
i1 @umulo.not.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL: umulo.not.i32:
; RV64:    # %bb.0: # %entry
; RV64-NEXT:  slli a1, a1, 32
; RV64-NEXT:  slli a0, a0, 32
; RV64-NEXT:  mulhu a0, a0, a1
; RV64-NEXT:  srai a0, a0, 32
; RV64-NEXT:  snez a0, a0
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.umul.with.overflow.i32(i32 %v1, i32 %v2)

```

```

%obit = extractvalue {i32, i1} %t, 1
%ret = xor i1 %obit, true
ret i1 %ret
}

define i64 @umulo.select.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: umulo.select.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  mulhu a2, a0, a1
; RV64-NEXT:  bnez a2, .LBB50_2
; RV64-NEXT:  # %bb.1: # %entry
; RV64-NEXT:  mv a0, a1
; RV64-NEXT:  .LBB50_2: # %entry
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.umul.with.overflow.i64(i64 %v1, i64 %v2)
%obit = extractvalue {i64, i1} %t, 1
%ret = select i1 %obit, i64 %v1, i64 %v2
ret i64 %ret
}

define i1 @umulo.not.i64(i64 %v1, i64 %v2)
{
; RV64-LABEL: umulo.not.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  mulhu a0, a0, a1
; RV64-NEXT:  snez a0, a0
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.umul.with.overflow.i64(i64 %v1, i64 %v2)
%obit = extractvalue {i64, i1} %t, 1
%ret = xor i1 %obit, true
ret i1 %ret
}

;
; Check the use of the overflow bit in combination with a branch instruction.
;
define zeroext i1 @saddo.br.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL: saddo.br.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addw a2, a0, a1
; RV64-NEXT:  slt a0, a2, a0
; RV64-NEXT:  slti a1, a1, 0
; RV64-NEXT:  beq a1, a0, .LBB52_2
; RV64-NEXT:  # %bb.1: # %overflow

```



```

; RV64-NEXT:  li a0, 0
; RV64-NEXT:  ret
; RV64-NEXT:  .LBB52_2: # %continue
; RV64-NEXT:  li a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.sadd.with.overflow.i32(i32 %v1, i32 %v2)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
br i1
%obit, label %overflow, label %continue

overflow:
ret i1 false

continue:
ret i1 true
}

```

```

define zeroext i1 @saddo.br.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: saddo.br.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  add a2, a0, a1
; RV64-NEXT:  slt a0, a2, a0
; RV64-NEXT:  slti a1, a1, 0
; RV64-NEXT:  beq a1, a0, .LBB53_2
; RV64-NEXT:  # %bb.1: # %overflow
; RV64-NEXT:  li a0, 0
; RV64-NEXT:  ret
; RV64-NEXT:  .LBB53_2: # %continue
; RV64-NEXT:  li a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.sadd.with.overflow.i64(i64 %v1, i64 %v2)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
br i1 %obit, label %overflow, label %continue

overflow:
ret i1 false

continue:
ret i1 true
}

```

```

define zeroext i1 @uaddo.br.i32(i32 %v1, i32 %v2) {
; RV64-LABEL: uaddo.br.i32:
; RV64:      # %bb.0: # %entry

```

```

; RV64-NEXT:   addw a1, a0, a1
; RV64-NEXT:   sext.w a0, a0
; RV64-NEXT:   bgeu a1, a0, .LBB54_2
; RV64-NEXT:   # %bb.1: # %overflow
;
RV64-NEXT:   li a0, 0
; RV64-NEXT:   ret
; RV64-NEXT:   .LBB54_2: # %continue
; RV64-NEXT:   li a0, 1
; RV64-NEXT:   ret
entry:
%t = call {i32, i1} @llvm.uadd.with.overflow.i32(i32 %v1, i32 %v2)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
br i1 %obit, label %overflow, label %continue

overflow:
ret i1 false

continue:
ret i1 true
}

define zeroext i1 @uaddo.br.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: uaddo.br.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:   add a1, a0, a1
; RV64-NEXT:   bgeu a1, a0, .LBB55_2
; RV64-NEXT:   # %bb.1: # %overflow
; RV64-NEXT:   li a0, 0
; RV64-NEXT:   ret
; RV64-NEXT:   .LBB55_2: # %continue
; RV64-NEXT:   li a0, 1
; RV64-NEXT:   ret
entry:
%t = call {i64, i1} @llvm.uadd.with.overflow.i64(i64 %v1, i64 %v2)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
br i1 %obit, label %overflow, label %continue

overflow:
ret i1 false

continue:
ret i1 true
}

define zeroext i1 @ssubo.br.i32(i32

```

```

    signext %v1, i32 signext %v2) {
; RV64-LABEL: ssubo.br.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  sgtz a2, a1
; RV64-NEXT:  subw a1, a0, a1
; RV64-NEXT:  slt a0, a1, a0
; RV64-NEXT:  beq a2, a0, .LBB56_2
; RV64-NEXT:  # %bb.1: # %overflow
; RV64-NEXT:  li a0, 0
; RV64-NEXT:  ret
; RV64-NEXT: .LBB56_2: # %continue
; RV64-NEXT:  li a0, 1
; RV64-NEXT:  ret
entry:
    %t = call {i32, i1} @llvm.ssub.with.overflow.i32(i32 %v1, i32 %v2)
    %val = extractvalue {i32, i1} %t, 0
    %obit = extractvalue {i32, i1} %t, 1
    br i1 %obit, label %overflow, label %continue

overflow:
    ret i1 false

continue:
    ret i1 true
}

```

```

define zeroext i1 @ssubo.br.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: ssubo.br.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  sgtz a2, a1
; RV64-NEXT:  sub a1, a0, a1
; RV64-NEXT:  slt a0, a1, a0
; RV64-NEXT:  beq a2, a0, .LBB57_2
; RV64-NEXT:  # %bb.1: # %overflow
; RV64-NEXT:  li a0, 0
; RV64-NEXT:  ret
; RV64-NEXT: .LBB57_2: # %continue
;
    RV64-NEXT:  li a0, 1
; RV64-NEXT:  ret
entry:
    %t = call {i64, i1} @llvm.ssub.with.overflow.i64(i64 %v1, i64 %v2)
    %val = extractvalue {i64, i1} %t, 0
    %obit = extractvalue {i64, i1} %t, 1
    br i1 %obit, label %overflow, label %continue

overflow:
    ret i1 false

```

```

continue:
    ret i1 true
}

define zeroext i1 @usubo.br.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL: usubo.br.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  subw a1, a0, a1
; RV64-NEXT:  bgeu a0, a1, .LBB58_2
; RV64-NEXT:  # %bb.1: # %overflow
; RV64-NEXT:  li a0, 0
; RV64-NEXT:  ret
; RV64-NEXT:  .LBB58_2: # %continue
; RV64-NEXT:  li a0, 1
; RV64-NEXT:  ret
entry:
    %t = call {i32, i1} @llvm.usub.with.overflow.i32(i32 %v1, i32 %v2)
    %val = extractvalue {i32, i1} %t, 0
    %obit = extractvalue {i32, i1} %t, 1
    br i1 %obit, label %overflow, label %continue

overflow:
    ret i1 false

continue:
    ret i1 true
}

define zeroext i1 @usubo.br.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: usubo.br.i64:
;
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  sub a1, a0, a1
; RV64-NEXT:  bgeu a0, a1, .LBB59_2
; RV64-NEXT:  # %bb.1: # %overflow
; RV64-NEXT:  li a0, 0
; RV64-NEXT:  ret
; RV64-NEXT:  .LBB59_2: # %continue
; RV64-NEXT:  li a0, 1
; RV64-NEXT:  ret
entry:
    %t = call {i64, i1} @llvm.usub.with.overflow.i64(i64 %v1, i64 %v2)
    %val = extractvalue {i64, i1} %t, 0
    %obit = extractvalue {i64, i1} %t, 1
    br i1 %obit, label %overflow, label %continue

overflow:

```

```

ret i1 false

continue:
ret i1 true
}

define zeroext i1 @smulo.br.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL: smulo.br.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  mul a0, a0, a1
; RV64-NEXT:  srai a1, a0, 32
; RV64-NEXT:  sraiw a0, a0, 31
; RV64-NEXT:  beq a1, a0, .LBB60_2
; RV64-NEXT:  # %bb.1: # %overflow
; RV64-NEXT:  li a0, 0
; RV64-NEXT:  ret
; RV64-NEXT: .LBB60_2: # %continue
; RV64-NEXT:  li a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.smul.with.overflow.i32(i32
%v1, i32 %v2)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
br i1 %obit, label %overflow, label %continue

overflow:
ret i1 false

continue:
ret i1 true
}

define zeroext i1 @smulo.br.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: smulo.br.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  mulh a2, a0, a1
; RV64-NEXT:  mul a0, a0, a1
; RV64-NEXT:  srai a0, a0, 63
; RV64-NEXT:  beq a2, a0, .LBB61_2
; RV64-NEXT:  # %bb.1: # %overflow
; RV64-NEXT:  li a0, 0
; RV64-NEXT:  ret
; RV64-NEXT: .LBB61_2: # %continue
; RV64-NEXT:  li a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.smul.with.overflow.i64(i64 %v1, i64 %v2)

```

```

%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
br i1 %obit, label %overflow, label %continue

```

```

overflow:
ret i1 false

```

```

continue:
ret i1 true
}

```

```

define zeroext i1 @smulo2.br.i64(i64 %v1) {
; RV64-LABEL: smulo2.br.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  li a1, -13
; RV64-NEXT:
    mulh a2, a0, a1
; RV64-NEXT:  mul a0, a0, a1
; RV64-NEXT:  srai a0, a0, 63
; RV64-NEXT:  beq a2, a0, .LBB62_2
; RV64-NEXT: # %bb.1: # %overflow
; RV64-NEXT:  li a0, 0
; RV64-NEXT:  ret
; RV64-NEXT: .LBB62_2: # %continue
; RV64-NEXT:  li a0, 1
; RV64-NEXT:  ret
entry:
    %t = call {i64, i1} @llvm.smul.with.overflow.i64(i64 %v1, i64 -13)
    %val = extractvalue {i64, i1} %t, 0
    %obit = extractvalue {i64, i1} %t, 1
    br i1 %obit, label %overflow, label %continue

```

```

overflow:
ret i1 false

```

```

continue:
ret i1 true
}

```

```

define zeroext i1 @umulo.br.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL: umulo.br.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  slli a1, a1, 32
; RV64-NEXT:  slli a0, a0, 32
; RV64-NEXT:  mulhu a0, a0, a1
; RV64-NEXT:  srai a0, a0, 32
; RV64-NEXT:  beqz a0, .LBB63_2
; RV64-NEXT: # %bb.1: # %overflow

```

```

; RV64-NEXT:  li a0, 0
; RV64-NEXT:  ret
; RV64-NEXT:  .LBB63_2: # %continue
; RV64-NEXT:  li a0, 1
; RV64-NEXT:  ret
entry:
%t = call
{i32, i1} @llvm.umul.with.overflow.i32(i32 %v1, i32 %v2)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
br i1 %obit, label %overflow, label %continue

overflow:
ret i1 false

continue:
ret i1 true
}

define zeroext i1 @umulo.br.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: umulo.br.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  mulhu a0, a0, a1
; RV64-NEXT:  beqz a0, .LBB64_2
; RV64-NEXT:  # %bb.1: # %overflow
; RV64-NEXT:  li a0, 0
; RV64-NEXT:  ret
; RV64-NEXT:  .LBB64_2: # %continue
; RV64-NEXT:  li a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.umul.with.overflow.i64(i64 %v1, i64 %v2)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
br i1 %obit, label %overflow, label %continue

overflow:
ret i1 false

continue:
ret i1 true
}

define zeroext i1 @umulo2.br.i64(i64 %v1) {
; RV64-LABEL: umulo2.br.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  add a1, a0, a0
; RV64-NEXT:  bgeu a1, a0,

```

```

.LBB65_2
; RV64-NEXT: # %bb.1: # %overflow
; RV64-NEXT: li a0, 0
; RV64-NEXT: ret
; RV64-NEXT: .LBB65_2: # %continue
; RV64-NEXT: li a0, 1
; RV64-NEXT: ret
entry:
%t = call {i64, i1} @llvm.umul.with.overflow.i64(i64 %v1, i64 2)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
br i1 %obit, label %overflow, label %continue

overflow:
ret i1 false

continue:
ret i1 true
}

define zeroext i1 @uaddo.i64.constant(i64 %v1, ptr %res) {
; RV64-LABEL: uaddo.i64.constant:
; RV64:    # %bb.0: # %entry
; RV64-NEXT: addi a2, a0, 2
; RV64-NEXT: sltu a0, a2, a0
; RV64-NEXT: sd a2, 0(a1)
; RV64-NEXT: ret
entry:
%t = call {i64, i1} @llvm.uadd.with.overflow.i64(i64 %v1, i64 2)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
store i64 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @uaddo.i64.constant_2048(i64 %v1, ptr %res) {
; RV64-LABEL: uaddo.i64.constant_2048:
; RV64:    # %bb.0: # %entry
; RV64-NEXT:
    addi a2, a0, 2047
; RV64-NEXT: addi a2, a2, 1
; RV64-NEXT: sltu a0, a2, a0
; RV64-NEXT: sd a2, 0(a1)
; RV64-NEXT: ret
entry:
%t = call {i64, i1} @llvm.uadd.with.overflow.i64(i64 %v1, i64 2048)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1

```



```

store i64 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @uaddo.i64.constant_2049(i64 %v1, ptr %res) {
; RV64-LABEL: uaddo.i64.constant_2049:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addi a2, a0, 2047
; RV64-NEXT:  addi a2, a2, 2
; RV64-NEXT:  sltu a0, a2, a0
; RV64-NEXT:  sd a2, 0(a1)
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.uadd.with.overflow.i64(i64 %v1, i64 2049)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
store i64 %val, ptr %res
ret i1 %obit
}

define i64 @uaddo.i64.constant_setcc_on_overflow_flag(ptr %p) {
; RV64-LABEL: uaddo.i64.constant_setcc_on_overflow_flag:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  ld a1, 0(a0)
;
; RV64-NEXT:  addi a0, a1, 2
; RV64-NEXT:  bltu a0, a1, .LBB69_2
; RV64-NEXT:  # %bb.1: # %IfOverflow
; RV64-NEXT:  li a0, 0
; RV64-NEXT:  .LBB69_2: # %IfNoOverflow
; RV64-NEXT:  ret
entry:
%v1 = load i64, ptr %p
%t = call {i64, i1} @llvm.uadd.with.overflow.i64(i64 %v1, i64 2)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
br i1 %obit, label %IfNoOverflow, label %IfOverflow
IfOverflow:
ret i64 0
IfNoOverflow:
ret i64 %val
}

declare {i32, i1} @llvm.sadd.with.overflow.i32(i32, i32) nounwind readnone
declare {i64, i1} @llvm.sadd.with.overflow.i64(i64, i64) nounwind readnone
declare {i32, i1} @llvm.uadd.with.overflow.i32(i32, i32) nounwind readnone
declare {i64, i1} @llvm.uadd.with.overflow.i64(i64, i64) nounwind readnone
declare {i32, i1} @llvm.ssub.with.overflow.i32(i32, i32) nounwind readnone

```

```
declare {i64, i1} @llvm.ssub.with.overflow.i64(i64, i64) nounwind readnone
declare {i32, i1} @llvm.usub.with.overflow.i32(i32, i32) nounwind readnone
declare
{i64, i1} @llvm.usub.with.overflow.i64(i64, i64) nounwind readnone
declare {i32, i1} @llvm.smul.with.overflow.i32(i32, i32) nounwind readnone
declare {i64, i1} @llvm.smul.with.overflow.i64(i64, i64) nounwind readnone
declare {i32, i1} @llvm.umul.with.overflow.i32(i32, i32) nounwind readnone
declare {i64, i1} @llvm.umul.with.overflow.i64(i64, i64) nounwind readnone
```

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use,

reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,

defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

### ---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

=====

Software from third parties included in the LLVM Project:

=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or `LICENSE` file at the top containing the specific license and restrictions which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every file.

=====

Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):

=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2007-2019 University of Illinois at Urbana-Champaign.  
All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

<http://llvm.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- \* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to

endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

```
add_lldb_library(lldbPluginObjectContainerBSDArchive PLUGIN
ObjectContainerBSDArchive.cpp
```

```
LINK_LIBS
```

```
  lldbCore
```

```
  lldbHost
```

```
  lldbSymbol
```

```
LINK_COMPONENTS
```

```
  Support
```

```
)
```

```
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
```

```
; RUN: llc -mtriple=riscv64 -verify-machineinstrs < %s \
```

```
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck -check-prefix=RV64I %s
```

```
; RUN: llc -mtriple=riscv64 -mattr=+m -verify-machineinstrs < %s \
```

```
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck -check-prefix=RV64IM %s
```

```
define i32 @udiv(i32 %a, i32 %b) nounwind {
```

```
; RV64I-LABEL: udiv:
```

```
; RV64I:      # %bb.0:
```

```
; RV64I-NEXT:  addi sp, sp, -16
```

```
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
```

```
; RV64I-NEXT:  slli a0, a0, 32
```

```
; RV64I-NEXT:  srli a0, a0, 32
```

```
; RV64I-NEXT:  slli a1, a1, 32
```

```
; RV64I-NEXT:  srli a1, a1, 32
```

```
; RV64I-NEXT:  call __udivdi3
```

```
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
```

```
; RV64I-NEXT:  addi sp, sp, 16
```

```
; RV64I-NEXT:  ret
```

```
;
```

```
; RV64IM-LABEL: udiv:
```

```
; RV64IM:      # %bb.0:
```

```
; RV64IM-NEXT:  divuw a0, a0, a1
```

```
; RV64IM-NEXT:  ret
```

```
%1 = udiv i32 %a, %b
```

```
ret i32 %1
```

```
}
```



```

define i32 @udiv_constant(i32 %a) nounwind {
; RV64I-LABEL:
    udiv_constant:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  li a1, 5
; RV64I-NEXT:  call __udivdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: udiv_constant:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  slli a0, a0, 32
; RV64IM-NEXT:  lui a1, 838861
; RV64IM-NEXT:  addi a1, a1, -819
; RV64IM-NEXT:  slli a1, a1, 32
; RV64IM-NEXT:  mulhu a0, a0, a1
; RV64IM-NEXT:  srli a0, a0, 34
; RV64IM-NEXT:  ret
    %1 = udiv i32 %a, 5
    ret i32 %1
}

```

```

define i32 @udiv_pow2(i32 %a) nounwind {
; RV64I-LABEL: udiv_pow2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srliw a0, a0, 3
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: udiv_pow2:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  srliw a0, a0, 3
; RV64IM-NEXT:  ret
    %1 = udiv i32 %a, 8
    ret i32 %1
}

```

```

define i32 @udiv_constant_lhs(i32
    %a) nounwind {
; RV64I-LABEL: udiv_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill

```

```

; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a1, a0, 32
; RV64I-NEXT:  li a0, 10
; RV64I-NEXT:  call __udivdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: udiv_constant_lhs:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  li a1, 10
; RV64IM-NEXT:  divuw a0, a1, a0
; RV64IM-NEXT:  ret
%1 = udiv i32 10, %a
ret i32 %1
}

```

```

define i64 @udiv64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: udiv64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  tail __udivdi3
;
; RV64IM-LABEL: udiv64:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  divu a0, a0, a1
; RV64IM-NEXT:  ret
%1 = udiv i64 %a, %b
ret i64 %1
}

```

```

define i64 @udiv64_constant(i64 %a) nounwind {
; RV64I-LABEL: udiv64_constant:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 5
; RV64I-NEXT:
;         tail __udivdi3
;
; RV64IM-LABEL: udiv64_constant:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  lui a1, 838861
; RV64IM-NEXT:  addiw a1, a1, -819
; RV64IM-NEXT:  slli a2, a1, 32
; RV64IM-NEXT:  add a1, a1, a2
; RV64IM-NEXT:  mulhu a0, a0, a1
; RV64IM-NEXT:  srli a0, a0, 2
; RV64IM-NEXT:  ret
%1 = udiv i64 %a, 5
ret i64 %1
}

```

```

}

define i64 @udiv64_constant_lhs(i64 %a) nounwind {
; RV64I-LABEL: udiv64_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  mv a1, a0
; RV64I-NEXT:  li a0, 10
; RV64I-NEXT:  tail __udivdi3
;
; RV64IM-LABEL: udiv64_constant_lhs:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  li a1, 10
; RV64IM-NEXT:  divu a0, a1, a0
; RV64IM-NEXT:  ret
%1 = udiv i64 10, %a
ret i64 %1
}

```

```

define i8 @udiv8(i8 %a, i8 %b) nounwind {
; RV64I-LABEL: udiv8:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  andi a0, a0, 255
; RV64I-NEXT:  andi a1, a1, 255
; RV64I-NEXT:  call
__udivdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: udiv8:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  andi a1, a1, 255
; RV64IM-NEXT:  andi a0, a0, 255
; RV64IM-NEXT:  divuw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = udiv i8 %a, %b
ret i8 %1
}

```

```

define i8 @udiv8_constant(i8 %a) nounwind {
; RV64I-LABEL: udiv8_constant:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  andi a0, a0, 255
; RV64I-NEXT:  li a1, 5

```

```

; RV64I-NEXT:  call __udivdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: udiv8_constant:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  andi a0, a0, 255
; RV64IM-NEXT:  li a1, 205
; RV64IM-NEXT:  mul a0, a0, a1
; RV64IM-NEXT:  srliw a0, a0, 10
; RV64IM-NEXT:  ret
%1 = udiv i8 %a, 5
ret i8 %1
}

```

```

define i8 @udiv8_pow2(i8 %a) nounwind
{
; RV64I-LABEL: udiv8_pow2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 56
; RV64I-NEXT:  srli a0, a0, 59
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: udiv8_pow2:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  slli a0, a0, 56
; RV64IM-NEXT:  srli a0, a0, 59
; RV64IM-NEXT:  ret
%1 = udiv i8 %a, 8
ret i8 %1
}

```

```

define i8 @udiv8_constant_lhs(i8 %a) nounwind {
; RV64I-LABEL: udiv8_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  andi a1, a0, 255
; RV64I-NEXT:  li a0, 10
; RV64I-NEXT:  call __udivdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: udiv8_constant_lhs:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  andi a0, a0, 255

```

```

; RV64IM-NEXT:  li a1, 10
; RV64IM-NEXT:  divuw a0, a1, a0
; RV64IM-NEXT:  ret
%1 = udiv i8 10, %a
ret i8 %1
}

define i16 @udiv16(i16 %a, i16 %b) nounwind {
; RV64I-LABEL: udiv16:
;
RV64I:    # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  lui a2, 16
; RV64I-NEXT:  addiw a2, a2, -1
; RV64I-NEXT:  and a0, a0, a2
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  call __udivdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: udiv16:
; RV64IM:    # %bb.0:
; RV64IM-NEXT:  lui a2, 16
; RV64IM-NEXT:  addi a2, a2, -1
; RV64IM-NEXT:  and a1, a1, a2
; RV64IM-NEXT:  and a0, a0, a2
; RV64IM-NEXT:  divuw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = udiv i16 %a, %b
ret i16 %1
}

```

```

define i16 @udiv16_constant(i16 %a) nounwind {
; RV64I-LABEL: udiv16_constant:
; RV64I:    # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srli a0, a0, 48
; RV64I-NEXT:  li a1, 5
; RV64I-NEXT:  call __udivdi3
; RV64I-NEXT:  ld ra, 8(sp)
# 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;

```

```

; RV64IM-LABEL: udiv16_constant:
; RV64IM:    # %bb.0:
; RV64IM-NEXT:  lui a1, 52429
; RV64IM-NEXT:  slli a1, a1, 4
; RV64IM-NEXT:  slli a0, a0, 48
; RV64IM-NEXT:  mulhu a0, a0, a1
; RV64IM-NEXT:  srliw a0, a0, 18
; RV64IM-NEXT:  ret
%1 = udiv i16 %a, 5
ret i16 %1
}

define i16 @udiv16_pow2(i16 %a) nounwind {
; RV64I-LABEL: udiv16_pow2:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srli a0, a0, 51
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: udiv16_pow2:
; RV64IM:    # %bb.0:
; RV64IM-NEXT:  slli a0, a0, 48
; RV64IM-NEXT:  srli a0, a0, 51
; RV64IM-NEXT:  ret
%1 = udiv i16 %a, 8
ret i16 %1
}

define i16 @udiv16_constant_lhs(i16 %a) nounwind {
; RV64I-LABEL: udiv16_constant_lhs:
; RV64I:    # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srli a1, a0, 48
; RV64I-NEXT:
    li a0, 10
; RV64I-NEXT:  call __udivdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: udiv16_constant_lhs:
; RV64IM:    # %bb.0:
; RV64IM-NEXT:  slli a0, a0, 48
; RV64IM-NEXT:  srli a0, a0, 48
; RV64IM-NEXT:  li a1, 10
; RV64IM-NEXT:  divuw a0, a1, a0

```

```

; RV64IM-NEXT:  ret
%1 = udiv i16 10, %a
ret i16 %1
}

define i32 @sdiv(i32 %a, i32 %b) nounwind {
; RV64I-LABEL: sdiv:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  sext.w a0, a0
; RV64I-NEXT:  sext.w a1, a1
; RV64I-NEXT:  call __divdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: sdiv:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  divw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = sdiv i32 %a, %b
ret i32 %1
}

define i32 @sdiv_constant(i32 %a) nounwind {
; RV64I-LABEL: sdiv_constant:
;
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  sext.w a0, a0
; RV64I-NEXT:  li a1, 5
; RV64I-NEXT:  call __divdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: sdiv_constant:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  sext.w a0, a0
; RV64IM-NEXT:  lui a1, 419430
; RV64IM-NEXT:  addiw a1, a1, 1639
; RV64IM-NEXT:  mul a0, a0, a1
; RV64IM-NEXT:  srli a1, a0, 63
; RV64IM-NEXT:  srai a0, a0, 33
; RV64IM-NEXT:  addw a0, a0, a1
; RV64IM-NEXT:  ret

```

```

%1 = sdiv i32 %a, 5
ret i32 %1
}

```

```

define i32 @sdiv_pow2(i32 %a) nounwind {
; RV64I-LABEL: sdiv_pow2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  sraiw a1, a0, 31
; RV64I-NEXT:  srliw a1, a1, 29
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  sraiw a0, a0, 3
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: sdiv_pow2:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  sraiw a1, a0,
31
; RV64IM-NEXT:  srliw a1, a1, 29
; RV64IM-NEXT:  add a0, a0, a1
; RV64IM-NEXT:  sraiw a0, a0, 3
; RV64IM-NEXT:  ret
%1 = sdiv i32 %a, 8
ret i32 %1
}

```

```

define i32 @sdiv_pow2_2(i32 %a) nounwind {
; RV64I-LABEL: sdiv_pow2_2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  sraiw a1, a0, 31
; RV64I-NEXT:  srliw a1, a1, 16
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  sraiw a0, a0, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: sdiv_pow2_2:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  sraiw a1, a0, 31
; RV64IM-NEXT:  srliw a1, a1, 16
; RV64IM-NEXT:  add a0, a0, a1
; RV64IM-NEXT:  sraiw a0, a0, 16
; RV64IM-NEXT:  ret
%1 = sdiv i32 %a, 65536
ret i32 %1
}

```

```

define i32 @sdiv_constant_lhs(i32 %a) nounwind {
; RV64I-LABEL: sdiv_constant_lhs:
; RV64I:      # %bb.0:

```



```

; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  sext.w a1, a0
; RV64I-NEXT:  li a0, -10
; RV64I-NEXT:  call __divdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded
Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: sdiv_constant_lhs:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  li a1, -10
; RV64IM-NEXT:  divw a0, a1, a0
; RV64IM-NEXT:  ret
%1 = sdiv i32 -10, %a
ret i32 %1
}

```

```

define i64 @sdiv64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: sdiv64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  tail __divdi3
;
; RV64IM-LABEL: sdiv64:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  div a0, a0, a1
; RV64IM-NEXT:  ret
%1 = sdiv i64 %a, %b
ret i64 %1
}

```

```

define i64 @sdiv64_constant(i64 %a) nounwind {
; RV64I-LABEL: sdiv64_constant:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 5
; RV64I-NEXT:  tail __divdi3
;
; RV64IM-LABEL: sdiv64_constant:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  lui a1, %hi(.LCPI21_0)
; RV64IM-NEXT:  ld a1, %lo(.LCPI21_0)(a1)
; RV64IM-NEXT:  mulh a0, a0, a1
; RV64IM-NEXT:  srli a1, a0, 63
; RV64IM-NEXT:  srai a0, a0, 1
; RV64IM-NEXT:  add a0, a0, a1
; RV64IM-NEXT:  ret
%1 = sdiv i64 %a,

```

5

```
ret i64 %1
}
```

```
define i64 @sdiv64_constant_lhs(i64 %a) nounwind {
; RV64I-LABEL: sdiv64_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  mv a1, a0
; RV64I-NEXT:  li a0, 10
; RV64I-NEXT:  tail __divdi3
;
; RV64IM-LABEL: sdiv64_constant_lhs:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  li a1, 10
; RV64IM-NEXT:  div a0, a1, a0
; RV64IM-NEXT:  ret
%1 = sdiv i64 10, %a
ret i64 %1
}
```

; Although this sdiv has two sexti32 operands, it shouldn't compile to divw on  
; RV64M as that wouldn't produce the correct result for e.g. INT\_MIN/-1.

```
define i64 @sdiv64_sext_operands(i32 %a, i32 %b) nounwind {
; RV64I-LABEL: sdiv64_sext_operands:
; RV64I:      # %bb.0:
; RV64I-NEXT:  sext.w a0, a0
; RV64I-NEXT:  sext.w a1, a1
; RV64I-NEXT:  tail __divdi3
;
; RV64IM-LABEL: sdiv64_sext_operands:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  sext.w a0, a0
; RV64IM-NEXT:  sext.w a1, a1
; RV64IM-NEXT:  div a0, a0, a1
; RV64IM-NEXT:  ret
%1 = sext i32 %a to i64
%2 = sext i32 %b to
i64
%3 = sdiv i64 %1, %2
ret i64 %3
}
```

```
define i8 @sdiv8(i8 %a, i8 %b) nounwind {
; RV64I-LABEL: sdiv8:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
```

```

; RV64I-NEXT:  slli a1, a1, 24
; RV64I-NEXT:  sraiw a1, a1, 24
; RV64I-NEXT:  slli a0, a0, 24
; RV64I-NEXT:  sraiw a0, a0, 24
; RV64I-NEXT:  call __divdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: sdiv8:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  slli a1, a1, 24
; RV64IM-NEXT:  sraiw a1, a1, 24
; RV64IM-NEXT:  slli a0, a0, 24
; RV64IM-NEXT:  sraiw a0, a0, 24
; RV64IM-NEXT:  divw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = sdiv i8 %a, %b
ret i8 %1
}

```

```

define i8 @sdiv8_constant(i8 %a) nounwind {
; RV64I-LABEL: sdiv8_constant:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a0, a0, 24
; RV64I-NEXT:
    sraiw a0, a0, 24
; RV64I-NEXT:  li a1, 5
; RV64I-NEXT:  call __divdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: sdiv8_constant:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  slli a0, a0, 24
; RV64IM-NEXT:  sraiw a0, a0, 24
; RV64IM-NEXT:  li a1, 103
; RV64IM-NEXT:  mul a0, a0, a1
; RV64IM-NEXT:  sraiw a1, a0, 9
; RV64IM-NEXT:  slli a0, a0, 48
; RV64IM-NEXT:  srli a0, a0, 63
; RV64IM-NEXT:  addw a0, a1, a0
; RV64IM-NEXT:  ret
%1 = sdiv i8 %a, 5
ret i8 %1
}

```

```
}
```

```
define i8 @sdiv8_pow2(i8 %a) nounwind {  
; RV64I-LABEL: sdiv8_pow2:  
; RV64I:      # %bb.0:  
; RV64I-NEXT: slli a1, a0, 24  
; RV64I-NEXT: sraiw a1, a1, 24  
; RV64I-NEXT: slli a1, a1, 49  
; RV64I-NEXT: srli a1, a1, 61  
; RV64I-NEXT: add a0, a0, a1  
; RV64I-NEXT: slli a0, a0, 24  
; RV64I-NEXT: sraiw a0, a0, 27  
; RV64I-NEXT: ret  
;  
; RV64IM-LABEL: sdiv8_pow2:  
; RV64IM:      # %bb.0:  
; RV64IM-NEXT: slli a1, a0,  
24  
; RV64IM-NEXT: sraiw a1, a1, 24  
; RV64IM-NEXT: slli a1, a1, 49  
; RV64IM-NEXT: srli a1, a1, 61  
; RV64IM-NEXT: add a0, a0, a1  
; RV64IM-NEXT: slli a0, a0, 24  
; RV64IM-NEXT: sraiw a0, a0, 27  
; RV64IM-NEXT: ret  
%1 = sdiv i8 %a, 8  
ret i8 %1  
}
```

```
define i8 @sdiv8_constant_lhs(i8 %a) nounwind {  
; RV64I-LABEL: sdiv8_constant_lhs:  
; RV64I:      # %bb.0:  
; RV64I-NEXT: addi sp, sp, -16  
; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill  
; RV64I-NEXT: slli a0, a0, 24  
; RV64I-NEXT: sraiw a1, a0, 24  
; RV64I-NEXT: li a0, -10  
; RV64I-NEXT: call __divdi3  
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload  
; RV64I-NEXT: addi sp, sp, 16  
; RV64I-NEXT: ret  
;  
; RV64IM-LABEL: sdiv8_constant_lhs:  
; RV64IM:      # %bb.0:  
; RV64IM-NEXT: slli a0, a0, 24  
; RV64IM-NEXT: sraiw a0, a0, 24  
; RV64IM-NEXT: li a1, -10  
; RV64IM-NEXT: divw a0, a1, a0
```

```

; RV64IM-NEXT:  ret
%1 = sdiv i8 -10, %a
ret i8 %1
}

define i16 @sdiv16(i16 %a, i16 %b) nounwind {
; RV64I-LABEL:
sdiv16:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a1, a1, 16
; RV64I-NEXT:  sraiw a1, a1, 16
; RV64I-NEXT:  slli a0, a0, 16
; RV64I-NEXT:  sraiw a0, a0, 16
; RV64I-NEXT:  call __divdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: sdiv16:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  slli a1, a1, 16
; RV64IM-NEXT:  sraiw a1, a1, 16
; RV64IM-NEXT:  slli a0, a0, 16
; RV64IM-NEXT:  sraiw a0, a0, 16
; RV64IM-NEXT:  divw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = sdiv i16 %a, %b
ret i16 %1
}

define i16 @sdiv16_constant(i16 %a) nounwind {
; RV64I-LABEL: sdiv16_constant:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a0, a0, 16
; RV64I-NEXT:  sraiw a0, a0, 16
; RV64I-NEXT:  li a1, 5
; RV64I-NEXT:  call __divdi3
; RV64I-NEXT:
    ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: sdiv16_constant:
; RV64IM:      # %bb.0:

```

```

; RV64IM-NEXT:  slli a0, a0, 16
; RV64IM-NEXT:  sraiw a0, a0, 16
; RV64IM-NEXT:  lui a1, 6
; RV64IM-NEXT:  addi a1, a1, 1639
; RV64IM-NEXT:  mul a0, a0, a1
; RV64IM-NEXT:  srliw a1, a0, 31
; RV64IM-NEXT:  sraiw a0, a0, 17
; RV64IM-NEXT:  addw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = sdiv i16 %a, 5
ret i16 %1
}

```

```

define i16 @sdiv16_pow2(i16 %a) nounwind {
; RV64I-LABEL: sdiv16_pow2:
; RV64I:      # %bb.0:
; RV64I-NEXT: slli a1, a0, 16
; RV64I-NEXT: sraiw a1, a1, 16
; RV64I-NEXT: slli a1, a1, 33
; RV64I-NEXT: srli a1, a1, 61
; RV64I-NEXT: add a0, a0, a1
; RV64I-NEXT: slli a0, a0, 16
; RV64I-NEXT: sraiw a0, a0, 19
; RV64I-NEXT: ret
;
; RV64IM-LABEL: sdiv16_pow2:
; RV64IM:     # %bb.0:
; RV64IM-NEXT: slli a1, a0, 16
; RV64IM-NEXT: sraiw a1, a1, 16
; RV64IM-NEXT: slli a1, a1, 33
;
; RV64IM-NEXT: srli a1, a1, 61
; RV64IM-NEXT: add a0, a0, a1
; RV64IM-NEXT: slli a0, a0, 16
; RV64IM-NEXT: sraiw a0, a0, 19
; RV64IM-NEXT: ret
%1 = sdiv i16 %a, 8
ret i16 %1
}

```

```

define i16 @sdiv16_constant_lhs(i16 %a) nounwind {
; RV64I-LABEL: sdiv16_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT: addi sp, sp, -16
; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT: slli a0, a0, 16
; RV64I-NEXT: sraiw a1, a0, 16
; RV64I-NEXT: li a0, -10

```

```

; RV64I-NEXT:  call __divdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: sdiv16_constant_lhs:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  slli a0, a0, 16
; RV64IM-NEXT:  sraiw a0, a0, 16
; RV64IM-NEXT:  li a1, -10
; RV64IM-NEXT:  divw a0, a1, a0
; RV64IM-NEXT:  ret
%1 = sdiv i16 -10, %a
ret i16 %1
}
LLVM System Interface Library
-----

```

Part of the LLVM Project, under the Apache License v2.0 with LLVM Exceptions.  
See <https://llvm.org/LICENSE.txt> for license information.  
SPDX-License-Identifier: Apache-2.0 WITH LLVM-exception

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable



copyright license to reproduce, prepare Derivative Works of,  
publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions

of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

=====

Software from third parties included in the LLVM Project:

=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or `LICENSE` file at the top containing the specific license and restrictions which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every file.

=====

Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):

=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2011-2019 by the contributors listed in CREDITS.TXT  
All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

<http://llvm.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies

of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- \* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative

Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be



liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

=====  
Software from third parties included in the LLVM Project:  
=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or `LICENSE` file at the top containing the specific license and restrictions which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every file.

=====  
Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):  
=====

The software contained in this directory tree is dual licensed under both the University of Illinois "BSD-Like" license and the MIT license. As a user of this code you may choose to use it under either license. As a contributor, you agree to allow your code to be used under both. The full text of the relevant licenses is included below.

In addition, a license agreement from the copyright/patent holders of the software contained in this directory tree is included below.

=====  
University of Illinois/NCSA  
Open Source License

Copyright  
(c) 1997-2019 Intel Corporation

All rights reserved.

Developed by:  
OpenMP Runtime Team  
Intel Corporation  
<http://www.openmp.rtl.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- \* Neither the names of Intel Corporation OpenMP Runtime Team nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====

Copyright (c) 1997-2019 Intel Corporation

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

Intel Corporation

Software Grant License Agreement ("Agreement")

Except for the license granted herein to you, Intel Corporation ("Intel") reserves all right, title, and interest in and to the Software (defined below).

Definition

"Software" means the code and documentation as well as any original work of authorship, including any modifications or additions to an existing work, that is intentionally submitted by Intel to [llvm.org](http://llvm.org) (<http://llvm.org>) ("LLVM") for inclusion in, or documentation of, any of the products owned or managed by LLVM (the "Work"). For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to LLVM or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, LLVM for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked otherwise.

1. Grant of Copyright License. Subject to the terms and conditions of this Agreement, Intel hereby grants to you and to recipients of the Software distributed by LLVM a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, sublicense, and distribute the Software and such derivative works.
2. Grant of Patent License. Subject to the terms and conditions of this Agreement, Intel hereby grants you and to recipients of the Software distributed by LLVM a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer

the Work, where such license applies only to those patent claims licensable by Intel that are necessarily infringed by Intel's Software alone or by combination of the Software with the Work to which such Software was submitted. If any entity institutes patent litigation against Intel or any other entity (including a cross-claim or counterclaim in a lawsuit) alleging that Intel's Software, or the Work to which Intel has contributed constitutes direct or contributory patent infringement, then any patent licenses granted to that entity under this Agreement for the Software or Work shall terminate as of the date such litigation is filed.

Unless required by applicable law or agreed to in writing, the software is provided on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.

=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.



9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition,

"submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each

Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions  
of this Software are embedded into an Object form of such source code, you  
may redistribute such embedded portions in such Object form without complying  
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled  
forms of this Software with  
software that is licensed under the GPLv2 ("Combined Software") and if a  
court of competent jurisdiction determines that the patent provision (Section  
3), the indemnity provision (Section 9) or other Section of the License  
conflicts with the conditions of the GPLv2, you may retroactively and  
prospectively choose to deem waived or otherwise exclude such Section(s) of  
the License, but only in their entirety and only with respect to the Combined  
Software.

=====  
Software from third parties included in the LLVM Project:  
=====

The LLVM Project contains third party software which is under different license  
terms. All such code will be identified clearly using at least one of two  
mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or  
`LICENSE` file at the top containing the specific  
license and restrictions  
which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every  
file.

=====  
Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):  
=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2009-2019 Polly Team  
All rights reserved.

Developed by:

Polly Team

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- \* Neither the names of the Polly Team, copyright holders, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to



communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

## APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

=====  
Software from third parties included in the LLVM Project:  
=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or `LICENSE` file at the top containing the specific license and restrictions which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every file.

=====  
Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):  
=====

The compiler\_rt library is dual licensed under both the University of Illinois "BSD-Like" license and the MIT license. As a user of this code you may choose to use it under either license. As a contributor, you agree to allow your code to be used under both.

Full text of the relevant licenses is included below.

=====

University of Illinois/NCSA

Open Source License

Copyright (c) 2009-2019 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

<http://llvm.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- \* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====  
Copyright (c) 2009-2015 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

; NOTE: Assertions have been autogenerated by utils/update\_llc\_test\_checks.py

; RUN: llc -mtriple=riscv64 -verify-machineinstrs < %s \

; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck -check-prefix=RV64I %s

; Check indexed and unindexed, sext, zext and anyext loads

```
define dso_local i64 @lb(ptr %a) nounwind {  
; RV64I-LABEL: lb:  
; RV64I:      # %bb.0:  
; RV64I-NEXT: lb a1, 1(a0)  
; RV64I-NEXT: lbu zero, 0(a0)  
; RV64I-NEXT: mv a0, a1  
; RV64I-NEXT: ret  
%1 = getelementptr i8, ptr %a, i32 1  
%2 = load i8, ptr %1  
%3 = sext i8 %2 to i64  
; the unused load will produce an anyext for selection  
%4 = load volatile i8, ptr %a  
ret i64 %3  
}
```

```
define dso_local i64 @lh(ptr %a) nounwind {  
; RV64I-LABEL: lh:  
; RV64I:      # %bb.0:  
; RV64I-NEXT: lh a1, 4(a0)
```

```

; RV64I-NEXT:  lh zero, 0(a0)
; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT:  ret
%1 = getelementptr i16, ptr %a, i32 2
%2 = load i16, ptr %1
%3 = sext i16 %2 to i64
; the unused load will produce an anyext for selection
%4 = load volatile i16, ptr %a
ret i64 %3
}

```

```

define dso_local i64 @lw(ptr %a) nounwind {
; RV64I-LABEL: lw:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lw a1, 12(a0)
; RV64I-NEXT:  lw zero, 0(a0)
; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT:  ret
%1 = getelementptr i32, ptr %a, i32 3
%2 = load i32, ptr %1
%3 = sext i32 %2 to i64
; the unused load will produce an anyext for selection
%4 = load volatile i32, ptr %a
ret i64 %3
}

```

```

define dso_local i64 @lbu(ptr %a) nounwind {
; RV64I-LABEL: lbu:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lbu a1, 4(a0)
; RV64I-NEXT:  lbu a0, 0(a0)
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  ret
%1 = getelementptr i8, ptr %a, i32 4
%2 = load i8, ptr %1
%3 = zext i8 %2 to i64
%4 = load volatile i8, ptr %a
%5 = zext i8 %4 to i64
%6 = add i64 %3, %5
ret i64 %6
}

```

```

define dso_local i64 @lhu(ptr %a) nounwind {
; RV64I-LABEL: lhu:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lhu a1, 10(a0)
; RV64I-NEXT:  lhu a0, 0(a0)
; RV64I-NEXT:

```

```

    add a0, a1, a0
; RV64I-NEXT:  ret
%1 = getelementptr i16, ptr %a, i32 5
%2 = load i16, ptr %1
%3 = zext i16 %2 to i64
%4 = load volatile i16, ptr %a
%5 = zext i16 %4 to i64
%6 = add i64 %3, %5
ret i64 %6
}

```

```

define dso_local i64 @lwu(ptr %a) nounwind {
; RV64I-LABEL: lwu:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lwu a1, 24(a0)
; RV64I-NEXT:  lwu a0, 0(a0)
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  ret
%1 = getelementptr i32, ptr %a, i32 6
%2 = load i32, ptr %1
%3 = zext i32 %2 to i64
%4 = load volatile i32, ptr %a
%5 = zext i32 %4 to i64
%6 = add i64 %3, %5
ret i64 %6
}

```

; 64-bit loads and stores

```

define dso_local i64 @ld(ptr %a) nounwind {
; RV64I-LABEL: ld:
; RV64I:      # %bb.0:
; RV64I-NEXT:  ld a1, 80(a0)
; RV64I-NEXT:  ld zero, 0(a0)
; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT:  ret
%1 = getelementptr i64, ptr %a, i32 10
%2 = load i64, ptr %1
%3 = load volatile i64, ptr %a
ret i64 %2
}

```

```

define dso_local void @sd(ptr %a, i64 %b)
nounwind {
; RV64I-LABEL: sd:
; RV64I:      # %bb.0:
; RV64I-NEXT:  sd a1, 0(a0)
; RV64I-NEXT:  sd a1, 88(a0)

```



```

; RV64I-NEXT:  ret
store i64 %b, ptr %a
%1 = getelementptr i64, ptr %a, i32 11
store i64 %b, ptr %1
ret void
}

; Check load and store to an i1 location
define dso_local i64 @load_sext_zext_anyext_i1(ptr %a) nounwind {
; RV64I-LABEL: load_sext_zext_anyext_i1:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lbu a1, 1(a0)
; RV64I-NEXT:  lbu a2, 2(a0)
; RV64I-NEXT:  lbu zero, 0(a0)
; RV64I-NEXT:  sub a0, a2, a1
; RV64I-NEXT:  ret
; sextload i1
%1 = getelementptr i1, ptr %a, i32 1
%2 = load i1, ptr %1
%3 = sext i1 %2 to i64
; zextload i1
%4 = getelementptr i1, ptr %a, i32 2
%5 = load i1, ptr %4
%6 = zext i1 %5 to i64
%7 = add i64 %3, %6
; extload i1 (anyext). Produced as the load is unused.
%8 = load volatile i1, ptr %a
ret i64 %7
}

define dso_local i16 @load_sext_zext_anyext_i1_i16(ptr %a) nounwind {
; RV64I-LABEL: load_sext_zext_anyext_i1_i16:
;
; RV64I:      # %bb.0:
; RV64I-NEXT:  lbu a1, 1(a0)
; RV64I-NEXT:  lbu a2, 2(a0)
; RV64I-NEXT:  lbu zero, 0(a0)
; RV64I-NEXT:  subw a0, a2, a1
; RV64I-NEXT:  ret
; sextload i1
%1 = getelementptr i1, ptr %a, i32 1
%2 = load i1, ptr %1
%3 = sext i1 %2 to i16
; zextload i1
%4 = getelementptr i1, ptr %a, i32 2
%5 = load i1, ptr %4
%6 = zext i1 %5 to i16
%7 = add i16 %3, %6

```

```

; extload i1 (anyext). Produced as the load is unused.
%8 = load volatile i1, ptr %a
ret i16 %7
}

; Check load and store to a global
@G = dso_local global i64 0

define dso_local i64 @ld_sd_global(i64 %a) nounwind {
; RV64I-LABEL: ld_sd_global:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a2, %hi(G)
; RV64I-NEXT:  ld a1, %lo(G)(a2)
; RV64I-NEXT:  addi a3, a2, %lo(G)
; RV64I-NEXT:  sd a0, %lo(G)(a2)
; RV64I-NEXT:  ld zero, 72(a3)
; RV64I-NEXT:  sd a0, 72(a3)
; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT:  ret
%1 = load volatile i64, ptr @G
store i64 %a, ptr @G
%2 = getelementptr
i64, ptr @G, i64 9
%3 = load volatile i64, ptr %2
store i64 %a, ptr %2
ret i64 %1
}

define i64 @lw_near_local(ptr %a) {
; RV64I-LABEL: lw_near_local:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi a0, a0, 2047
; RV64I-NEXT:  ld a0, 9(a0)
; RV64I-NEXT:  ret
%1 = getelementptr inbounds i64, ptr %a, i64 257
%2 = load volatile i64, ptr %1
ret i64 %2
}

define void @st_near_local(ptr %a, i64 %b) {
; RV64I-LABEL: st_near_local:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi a0, a0, 2047
; RV64I-NEXT:  sd a1, 9(a0)
; RV64I-NEXT:  ret
%1 = getelementptr inbounds i64, ptr %a, i64 257
store i64 %b, ptr %1
ret void

```

```

}

define i64 @lw_sw_near_local(ptr %a, i64 %b) {
; RV64I-LABEL: lw_sw_near_local:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi a2, a0, 2047
; RV64I-NEXT:  ld a0, 9(a2)
; RV64I-NEXT:  sd a1, 9(a2)
; RV64I-NEXT:  ret
%1 = getelementptr inbounds i64, ptr %a, i64 257
%2 = load volatile i64, ptr %1
store i64 %b, ptr %1
ret i64 %2
}

define
i64 @lw_far_local(ptr %a) {
; RV64I-LABEL: lw_far_local:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a1, 8
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ld a0, -8(a0)
; RV64I-NEXT:  ret
%1 = getelementptr inbounds i64, ptr %a, i64 4095
%2 = load volatile i64, ptr %1
ret i64 %2
}

define void @st_far_local(ptr %a, i64 %b) {
; RV64I-LABEL: st_far_local:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a2, 8
; RV64I-NEXT:  add a0, a0, a2
; RV64I-NEXT:  sd a1, -8(a0)
; RV64I-NEXT:  ret
%1 = getelementptr inbounds i64, ptr %a, i64 4095
store i64 %b, ptr %1
ret void
}

define i64 @lw_sw_far_local(ptr %a, i64 %b) {
; RV64I-LABEL: lw_sw_far_local:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a2, 8
; RV64I-NEXT:  add a2, a0, a2
; RV64I-NEXT:  ld a0, -8(a2)
; RV64I-NEXT:  sd a1, -8(a2)
; RV64I-NEXT:  ret

```

```

%1 = getelementptr inbounds i64, ptr %a, i64 4095
%2 = load volatile i64, ptr %1
store i64 %b, ptr %1
ret i64 %2
}

```

```

; Make sure we don't fold the addiw
into the load offset. The sign extend of the
; addiw is required.
define i64 @lw_really_far_local(ptr %a) {
; RV64I-LABEL: lw_really_far_local:
; RV64I:      # %bb.0:
; RV64I-NEXT: lui a1, 524288
; RV64I-NEXT: addiw a1, a1, -2048
; RV64I-NEXT: add a0, a0, a1
; RV64I-NEXT: ld a0, 0(a0)
; RV64I-NEXT: ret
%1 = getelementptr inbounds i64, ptr %a, i64 268435200
%2 = load volatile i64, ptr %1
ret i64 %2
}

```

```

; Make sure we don't fold the addiw into the store offset. The sign extend of
; the addiw is required.
define void @st_really_far_local(ptr %a, i64 %b) {
; RV64I-LABEL: st_really_far_local:
; RV64I:      # %bb.0:
; RV64I-NEXT: lui a2, 524288
; RV64I-NEXT: addiw a2, a2, -2048
; RV64I-NEXT: add a0, a0, a2
; RV64I-NEXT: sd a1, 0(a0)
; RV64I-NEXT: ret
%1 = getelementptr inbounds i64, ptr %a, i64 268435200
store i64 %b, ptr %1
ret void
}

```

```

; Make sure we don't fold the addiw into the load/store offset. The sign extend
; of the addiw is required.
define
i64 @lw_sw_really_far_local(ptr %a, i64 %b) {
; RV64I-LABEL: lw_sw_really_far_local:
; RV64I:      # %bb.0:
; RV64I-NEXT: lui a2, 524288
; RV64I-NEXT: addiw a2, a2, -2048
; RV64I-NEXT: add a2, a0, a2
; RV64I-NEXT: ld a0, 0(a2)
; RV64I-NEXT: sd a1, 0(a2)

```

```

; RV64I-NEXT:  ret
%1 = getelementptr inbounds i64, ptr %a, i64 268435200
%2 = load volatile i64, ptr %1
store i64 %b, ptr %1
ret i64 %2
}

%struct.quux = type { i32, [0 x i8] }

; Make sure we don't remove the addi and fold the C from
; (add (addi FrameIndex, C), X) into the store address.
; FrameIndex cannot be the operand of an ADD. We must keep the ADDI.
define void @addi_fold_crash(i64 %arg) nounwind {
; RV64I-LABEL: addi_fold_crash:
; RV64I:      # %bb.0: # %bb
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  addi a1, sp, 4
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  sb zero, 0(a0)
; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT:  call snork
; RV64I-NEXT:
    ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
bb:
%tmp = alloca %struct.quux, align 8
%tmp1 = getelementptr inbounds %struct.quux, ptr %tmp, i64 0, i32 1
%tmp2 = getelementptr inbounds %struct.quux, ptr %tmp, i64 0, i32 1, i64 %arg
store i8 0, ptr %tmp2, align 1
call void @snork(ptr %tmp1)
ret void
}

declare void @snork(ptr)
; NOTE: Assertions have been autogenerated by utils/update_test_checks.py UTC_ARGS: --function-signature --
scrub-attributes
; RUN: opt -S -passes=argpromotion < %s | FileCheck %s
; Test that we only promote arguments when the caller/callee have compatible
; function attributes.

target triple = "x86_64-unknown-linux-gnu"

; This should promote
define internal fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512(ptr %arg, ptr
readonly %arg1) #0 {
; CHECK-LABEL: define {{{^@}}} @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512

```

```

; CHECK-SAME: (ptr [[ARG:%.*]], <8 x i64> [[ARG1_VAL:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: store <8 x i64> [[ARG1_VAL]], ptr [[ARG]]
; CHECK-NEXT: ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

define void @avx512_legal512_prefer512_call_avx512_legal512_prefer512(ptr %arg) #0 {
; CHECK-LABEL: define {{{^@}+}}@avx512_legal512_prefer512_call_avx512_legal512_prefer512
; CHECK-SAME:
(ptr [[ARG:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: [[TMP2:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: call void @llvm.memset.p0.i64(ptr align 32 [[TMP]], i8 0, i64 32, i1 false)
; CHECK-NEXT: [[TMP_VAL:%.*]] = load <8 x i64>, ptr [[TMP]]
; CHECK-NEXT: call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512(ptr
[[TMP2]], <8 x i64> [[TMP_VAL]])
; CHECK-NEXT: [[TMP4:%.*]] = load <8 x i64>, ptr [[TMP2]], align 32
; CHECK-NEXT: store <8 x i64> [[TMP4]], ptr [[ARG]], align 2
; CHECK-NEXT: ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align 32
call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512(ptr %tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

; This should promote
define
internal fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256(ptr %arg, ptr readonly
%arg1) #1 {
; CHECK-LABEL: define {{{^@}+}}@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]], <8 x i64> [[ARG1_VAL:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: store <8 x i64> [[ARG1_VAL]], ptr [[ARG]]
; CHECK-NEXT: ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1

```

```

store <8 x i64> %tmp, ptr %arg
ret void
}

define void @avx512_legal512_prefer256_call_avx512_legal512_prefer256(ptr %arg) #1 {
; CHECK-LABEL: define {{{^@}+}}@avx512_legal512_prefer256_call_avx512_legal512_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: [[TMP2:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: call void @llvm.memset.p0.i64(ptr align 32 [[TMP]], i8 0, i64 32, i1 false)
; CHECK-NEXT: [[TMP_VAL:%.*]] = load <8 x i64>, ptr [[TMP]]
; CHECK-NEXT: call fastcc
void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256(ptr [[TMP2]], <8 x i64>
[[TMP_VAL]])
; CHECK-NEXT: [[TMP4:%.*]] = load <8 x i64>, ptr [[TMP2]], align 32
; CHECK-NEXT: store <8 x i64> [[TMP4]], ptr [[ARG]], align 2
; CHECK-NEXT: ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align 32
call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256(ptr %tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

; This should promote
define internal fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256(ptr %arg, ptr
readonly %arg1) #1 {
; CHECK-LABEL: define {{{^@}+}}@callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]], <8 x i64> [[ARG1_VAL:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: store <8 x i64> [[ARG1_VAL]],
ptr [[ARG]]
; CHECK-NEXT: ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

define void @avx512_legal512_prefer512_call_avx512_legal512_prefer256(ptr %arg) #0 {
; CHECK-LABEL: define {{{^@}+}}@avx512_legal512_prefer512_call_avx512_legal512_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]])

```

```

; CHECK-NEXT: bb:
; CHECK-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: [[TMP2:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: call void @llvm.memset.p0.i64(ptr align 32 [[TMP]], i8 0, i64 32, i1 false)
; CHECK-NEXT: [[TMP_VAL:%.*]] = load <8 x i64>, ptr [[TMP]]
; CHECK-NEXT: call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256(ptr
[[TMP2]], <8 x i64> [[TMP_VAL]])
; CHECK-NEXT: [[TMP4:%.*]] = load <8 x i64>, ptr [[TMP2]], align 32
; CHECK-NEXT: store <8 x i64> [[TMP4]], ptr [[ARG]], align 2
; CHECK-NEXT: ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align
32
call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256(ptr %tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

; This should promote
define internal fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512(ptr %arg, ptr
readonly %arg1) #0 {
; CHECK-LABEL: define {{^[^@]+}}@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512
; CHECK-SAME: (ptr [[ARG:%.*]], <8 x i64> [[ARG1_VAL:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: store <8 x i64> [[ARG1_VAL]], ptr [[ARG]]
; CHECK-NEXT: ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

define void @avx512_legal512_prefer256_call_avx512_legal512_prefer512(ptr %arg) #1 {
; CHECK-LABEL: define {{^[^@]+}}@avx512_legal512_prefer256_call_avx512_legal512_prefer512
; CHECK-SAME: (ptr [[ARG:%.*]])
; CHECK-NEXT:
bb:
; CHECK-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: [[TMP2:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: call void @llvm.memset.p0.i64(ptr align 32 [[TMP]], i8 0, i64 32, i1 false)
; CHECK-NEXT: [[TMP_VAL:%.*]] = load <8 x i64>, ptr [[TMP]]
; CHECK-NEXT: call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512(ptr
[[TMP2]], <8 x i64> [[TMP_VAL]])

```



```

; CHECK-NEXT: [[TMP4:%.*]] = load <8 x i64>, ptr [[TMP2]], align 32
; CHECK-NEXT: store <8 x i64> [[TMP4]], ptr [[ARG]], align 2
; CHECK-NEXT: ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align 32
call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512(ptr %tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

; This should not promote
define internal fastcc void @callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr
%arg, ptr readonly %arg1) #1 {
; CHECK-LABEL: define { {^@}+ } }@callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]], ptr readonly [[ARG1:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: [[TMP:%.*]] = load <8 x i64>, ptr [[ARG1]]
; CHECK-NEXT: store <8 x i64> [[TMP]], ptr [[ARG]]
; CHECK-NEXT: ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

define void @avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr %arg) #2 {
; CHECK-LABEL: define { {^@}+ } }@avx512_legal256_prefer256_call_avx512_legal512_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: [[TMP2:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: call void @llvm.memset.p0.i64(ptr align 32 [[TMP]], i8 0, i64 32, i1 false)
; CHECK-NEXT: call fastcc void @callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr
[[TMP2]], ptr [[TMP]])
; CHECK-NEXT: [[TMP4:%.*]] = load <8 x i64>, ptr [[TMP2]], align 32
; CHECK-NEXT: store <8 x i64> [[TMP4]], ptr [[ARG]], align 2
; CHECK-NEXT: ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align 32
call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
call fastcc void @callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr %tmp2, ptr %tmp)

```

```

%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

; This should not promote
define internal fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr %arg, ptr
readonly %arg1) #2 {
; CHECK-LABEL: define {{{^@+}}}@callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]], ptr readonly [[ARG1:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: [[TMP:%.*]] = load <8 x i64>, ptr [[ARG1]]
; CHECK-NEXT: store <8 x i64>
[[TMP]], ptr [[ARG]]
; CHECK-NEXT: ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

define void @avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr %arg) #1 {
; CHECK-LABEL: define {{{^@+}}}@avx512_legal512_prefer256_call_avx512_legal256_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: [[TMP2:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: call void @llvm.memset.p0.i64(ptr align 32 [[TMP]], i8 0, i64 32, i1 false)
; CHECK-NEXT: call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr
[[TMP2]], ptr [[TMP]])
; CHECK-NEXT: [[TMP4:%.*]] = load <8 x i64>, ptr [[TMP2]], align 32
; CHECK-NEXT: store <8 x i64> [[TMP4]], ptr [[ARG]], align 2
; CHECK-NEXT: ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align 32
call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64
32, i1 false)
call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr %tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

; This should promote
define internal fastcc void @callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256(ptr %arg, ptr readonly

```

```

%arg1) #3 {
; CHECK-LABEL: define {{^@}+}@callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]], <8 x i64> [[ARG1_VAL:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT:   store <8 x i64> [[ARG1_VAL]], ptr [[ARG]]
; CHECK-NEXT:   ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

define void @avx2_legal256_prefer256_call_avx2_legal512_prefer256(ptr %arg) #4 {
; CHECK-LABEL: define {{^@}+}@avx2_legal256_prefer256_call_avx2_legal512_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT:   [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT:
; CHECK-NEXT:   [[TMP2:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT:   call void @llvm.memset.p0.i64(ptr align 32 [[TMP]], i8 0, i64 32, i1 false)
; CHECK-NEXT:   [[TMP_VAL:%.*]] = load <8 x i64>, ptr [[TMP]]
; CHECK-NEXT:   call fastcc void @callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256(ptr
[[TMP2]], <8 x i64> [[TMP_VAL]])
; CHECK-NEXT:   [[TMP4:%.*]] = load <8 x i64>, ptr [[TMP2]], align 32
; CHECK-NEXT:   store <8 x i64> [[TMP4]], ptr [[ARG]], align 2
; CHECK-NEXT:   ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align 32
call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
call fastcc void @callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256(ptr %tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

; This should promote
define internal fastcc void @callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256(ptr %arg, ptr readonly
%arg1) #4 {
;
; CHECK-LABEL: define {{^@}+}@callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]], <8 x i64> [[ARG1_VAL:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT:   store <8 x i64> [[ARG1_VAL]], ptr [[ARG]]
; CHECK-NEXT:   ret void
;

```

```

bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

define void @avx2_legal512_prefer256_call_avx2_legal256_prefer256(ptr %arg) #3 {
; CHECK-LABEL: define {[^@]+}@avx2_legal512_prefer256_call_avx2_legal256_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: [[TMP2:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: call void @llvm.memset.p0.i64(ptr align 32 [[TMP]], i8 0, i64 32, i1 false)
; CHECK-NEXT: [[TMP_VAL:%.*]] = load <8 x i64>, ptr [[TMP]]
; CHECK-NEXT: call fastcc void @callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256(ptr
[[TMP2]], <8 x i64> [[TMP_VAL]])
; CHECK-NEXT: [[TMP4:%.*]] = load
<8 x i64>, ptr [[TMP2]], align 32
; CHECK-NEXT: store <8 x i64> [[TMP4]], ptr [[ARG]], align 2
; CHECK-NEXT: ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align 32
call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
call fastcc void @callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256(ptr %tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

; If the arguments are scalar, its ok to promote.
define internal i32 @scalar_callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr %X, ptr %Y)
#2 {
; CHECK-LABEL: define
{[ ^@ ]+}@scalar_callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256
; CHECK-SAME: (i32 [[X_VAL:%.*]], i32 [[Y_VAL:%.*]])
; CHECK-NEXT: [[C:%.*]] = add i32 [[X_VAL]], [[Y_VAL]]
; CHECK-NEXT: ret i32 [[C]]
;
%A = load i32, ptr %X
%B = load i32, ptr %Y
%C = add i32 %A, %B
ret i32 %C
}

define i32 @scalar_avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr

```

```

%B) #2 {
; CHECK-LABEL: define {{{^@}+}}@scalar_avx512_legal256_prefer256_call_avx512_legal512_prefer256
; CHECK-SAME: (ptr [[B:%.*]])
; CHECK-NEXT:  [[A:%.*]] = alloca i32
; CHECK-NEXT:  store i32 1, ptr [[A]]
; CHECK-NEXT:  [[A_VAL:%.*]] = load i32, ptr [[A]]
; CHECK-NEXT:  [[B_VAL:%.*]] = load i32, ptr [[B]]
; CHECK-NEXT:  [[C:%.*]] = call i32
@scalar_callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(i32 [[A_VAL]], i32 [[B_VAL]])
; CHECK-NEXT:  ret i32 [[C]]
;
%A = alloca i32
store i32 1, ptr %A
%C = call i32 @scalar_callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr %A, ptr %B)
ret i32 %C
}

```

; If the arguments are scalar, its ok to promote.

```

define internal i32 @scalar_callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr %X, ptr %Y)
#2 {
; CHECK-LABEL: define
{{{^@}+}}@scalar_callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256
; CHECK-SAME: (i32 [[X_VAL:%.*]],
i32 [[Y_VAL:%.*]])
; CHECK-NEXT:  [[C:%.*]] = add i32 [[X_VAL]], [[Y_VAL]]
; CHECK-NEXT:  ret i32 [[C]]
;
%A = load i32, ptr %X
%B = load i32, ptr %Y
%C = add i32 %A, %B
ret i32 %C
}

```

```

define i32 @scalar_avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr %B) #2 {
; CHECK-LABEL: define {{{^@}+}}@scalar_avx512_legal512_prefer256_call_avx512_legal256_prefer256
; CHECK-SAME: (ptr [[B:%.*]])
; CHECK-NEXT:  [[A:%.*]] = alloca i32
; CHECK-NEXT:  store i32 1, ptr [[A]]
; CHECK-NEXT:  [[A_VAL:%.*]] = load i32, ptr [[A]]
; CHECK-NEXT:  [[B_VAL:%.*]] = load i32, ptr [[B]]
; CHECK-NEXT:  [[C:%.*]] = call i32
@scalar_callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(i32 [[A_VAL]], i32 [[B_VAL]])
; CHECK-NEXT:  ret i32 [[C]]
;
%A = alloca i32
store i32 1, ptr %A
%C = call i32 @scalar_callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr %A, ptr %B)
ret i32 %C

```

```

}

; Function Attrs: argmemonly nounwind
declare void @llvm.memset.p0.i64(ptr
    nocapture writeonly, i8, i64, i1) #5

attributes #0 = { inlinehint norecurse nounwind uwtable "target-features"="+avx512vl" "min-legal-vector-
width"="512" "prefer-vector-width"="512" }
attributes #1 = { inlinehint norecurse nounwind uwtable "target-features"="+avx512vl" "min-legal-vector-
width"="512" "prefer-vector-width"="256" }
attributes #2 = { inlinehint norecurse nounwind uwtable "target-features"="+avx512vl" "min-legal-vector-
width"="256" "prefer-vector-width"="256" }
attributes #3 = { inlinehint norecurse nounwind uwtable "target-features"="+avx2" "min-legal-vector-width"="512"
"prefer-vector-width"="256" }
attributes #4 = { inlinehint norecurse nounwind uwtable "target-features"="+avx2" "min-legal-vector-width"="256"
"prefer-vector-width"="256" }
attributes #5 = { argmemonly nounwind }
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -mattr=+zbc -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64ZBC-ZBKC
; RUN: llc -mtriple=riscv64 -mattr=+zbkbc -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64ZBC-ZBKC

declare i64 @llvm.riscv.clmul.i64(i64 %a, i64 %b)

define i64 @clmul64(i64 %a, i64 %b) nounwind {
; RV64ZBC-ZBKC-LABEL: clmul64:
; RV64ZBC-ZBKC:      # %bb.0:
; RV64ZBC-ZBKC-NEXT:  clmul a0, a0, a1
; RV64ZBC-ZBKC-NEXT:  ret
    %tmp = call i64 @llvm.riscv.clmul.i64(i64 %a, i64 %b)
    ret i64 %tmp
}

declare i64 @llvm.riscv.clmulh.i64(i64 %a, i64 %b)

define i64 @clmul64h(i64 %a, i64 %b) nounwind {
; RV64ZBC-ZBKC-LABEL: clmul64h:
; RV64ZBC-ZBKC:      # %bb.0:
; RV64ZBC-ZBKC-NEXT:  clmulh a0, a0, a1
; RV64ZBC-ZBKC-NEXT:  ret
    %tmp = call i64 @llvm.riscv.clmulh.i64(i64 %a, i64 %b)
    ret i64 %tmp
}

declare
i32 @llvm.riscv.clmul.i32(i32 %a, i32 %b)

```

```

define signext i32 @clmul32(i32 signext %a, i32 signext %b) nounwind {
; RV64ZBC-ZBKC-LABEL: clmul32:
; RV64ZBC-ZBKC:      # %bb.0:
; RV64ZBC-ZBKC-NEXT:  clmul a0, a0, a1
; RV64ZBC-ZBKC-NEXT:  sext.w a0, a0
; RV64ZBC-ZBKC-NEXT:  ret
%tmp = call i32 @llvm.riscv.clmul.i32(i32 %a, i32 %b)
ret i32 %tmp
}

declare i32 @llvm.riscv.clmulh.i32(i32 %a, i32 %b)

define signext i32 @clmul32h(i32 signext %a, i32 signext %b) nounwind {
; RV64ZBC-ZBKC-LABEL: clmul32h:
; RV64ZBC-ZBKC:      # %bb.0:
; RV64ZBC-ZBKC-NEXT:  slli a1, a1, 32
; RV64ZBC-ZBKC-NEXT:  slli a0, a0, 32
; RV64ZBC-ZBKC-NEXT:  clmulh a0, a0, a1
; RV64ZBC-ZBKC-NEXT:  srai a0, a0, 32
; RV64ZBC-ZBKC-NEXT:  ret
%tmp = call i32 @llvm.riscv.clmulh.i32(i32 %a, i32 %b)
ret i32 %tmp
}

; FIXME: We could avoid the slli instructions by using clmul+srai since the
; inputs are zero extended.
define signext i32 @clmul32h_zext(i32 zeroext %a, i32 zeroext %b) nounwind {
;
; RV64ZBC-ZBKC-LABEL: clmul32h_zext:
; RV64ZBC-ZBKC:      # %bb.0:
; RV64ZBC-ZBKC-NEXT:  slli a1, a1, 32
; RV64ZBC-ZBKC-NEXT:  slli a0, a0, 32
; RV64ZBC-ZBKC-NEXT:  clmulh a0, a0, a1
; RV64ZBC-ZBKC-NEXT:  srai a0, a0, 32
; RV64ZBC-ZBKC-NEXT:  ret
%tmp = call i32 @llvm.riscv.clmulh.i32(i32 %a, i32 %b)
ret i32 %tmp
}

```

# 1.161 libssh 0.11.2-r0

## 1.161.1 Available under license :

MIT License

Copyright (c) 2017 Jan-Lukas Wynen

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get



it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the

entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

#### GNU LESSER GENERAL PUBLIC LICENSE

#### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves,

then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so

that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License.

Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data

structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

- c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any

attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people



have made  
generous contributions to the wide range of software distributed  
through that system in reliance on consistent application of that  
system; it is up to the author/donor to decide if he or she is willing  
to distribute software through any other system and a licensee cannot  
impose that choice.

This section is intended to make thoroughly clear what is believed to  
be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in  
certain countries either by patents or by copyrighted interfaces, the  
original copyright holder who places the Library under this License may add  
an explicit geographical distribution limitation excluding those countries,  
so that distribution is permitted only in or among countries not thus  
excluded. In such case, this License incorporates the limitation as if  
written in the body of this License.

13. The Free Software Foundation may publish revised and/or new  
versions of the Lesser General Public License  
from time to time.  
Such new versions will be similar in spirit to the present version,  
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library  
specifies a version number of this License which applies to it and  
"any later version", you have the option of following the terms and  
conditions either of that version or of any later version published by  
the Free Software Foundation. If the Library does not specify a  
license version number, you may choose any version ever published by  
the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free  
programs whose distribution conditions are incompatible with these,  
write to the author to ask for permission. For software which is  
copyrighted by the Free Software Foundation, write to the Free  
Software Foundation; we sometimes make exceptions for this. Our  
decision will be guided by the two goals of preserving the free status  
of  
all derivatives of our free software and of promoting the sharing  
and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO  
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.  
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR  
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY

KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL

DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### Linking with OpenSSL

17. In addition, as a special exception, we give permission to link the code of its release of libssh with the OpenSSL project's "OpenSSL" library (or with modified versions of it that use the same license as the "OpenSSL" library), and distribute the linked executables. You must obey the GNU Lesser General Public License in all respects for all of the code used other than "OpenSSL". If you modify this file, you may extend this exception to your version of the file, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

#### END OF TERMS AND CONDITIONS

Some parts are under the BSDv2 License :

Copyright (c) 2000 Markus Friedl. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT

NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;  
LOSS OF USE,  
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY  
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT  
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF  
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions  
are met:

1. Redistributions of source code must retain the copyright  
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the copyright  
notice, this list of conditions and the following disclaimer in the  
documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products  
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR  
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES  
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.  
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,  
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT  
NOT LIMITED TO, PROCUREMENT  
OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,  
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY  
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT  
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF  
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.162 tcpdump 4.99.5-r1

### 1.162.1 Available under license :

License: BSD

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions  
are met:

1. Redistributions of source code must retain the above copyright  
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright  
notice, this list of conditions and the following disclaimer in  
the documentation and/or other materials provided with the  
distribution.
3. The names of the authors may not be used to endorse or promote  
products derived from this software without specific prior

written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

```
# -*- perl -*-
```

```
$testlist = [  
  # This specific test fails on OpenBSD because the .pcap file uses DLT_RAW,  
  # which OpenBSD treats as DLT_LOOP.  
  {  
    config_unset => 'IS_OPENBSD',  
    name => 'heap-overflow-1',  
    input => 'heap-overflow-1.pcap',  
    output => 'heap-overflow-1.out',  
    args => '-v'  
  },  
];  
  
1;
```

## 1.163 python 3.12.12-r0

### 1.163.1 Available under license :

Additional Conditions for this Windows binary build

-----  
This program is linked with and uses Microsoft Distributable Code, copyrighted by Microsoft Corporation. The Microsoft Distributable Code is embedded in each .exe, .dll and .pyd file as a result of running the code through a linker.

If you further distribute programs that include the Microsoft Distributable Code, you must comply with the restrictions on distribution specified by Microsoft. In particular, you must require distributors and external end users to agree to terms that protect the Microsoft Distributable Code at least as much as Microsoft's own requirements for the Distributable Code. See Microsoft's documentation (included in its developer tools and on its website at [microsoft.com](https://www.microsoft.com)) for specific details.

Redistribution of the Windows binary build of the Python interpreter complies with this agreement, provided that you do not:

- alter any copyright, trademark or patent

notice in Microsoft's  
Distributable Code;

- use Microsoft's trademarks in your programs' names or in a way that suggests your programs come from or are endorsed by Microsoft;
- distribute Microsoft's Distributable Code to run on a platform other than Microsoft operating systems, run-time technologies or application platforms; or
- include Microsoft Distributable Code in malicious, deceptive or unlawful programs.

These restrictions apply only to the Microsoft Distributable Code as defined above, not to Python itself or any programs running on the Python interpreter. The redistribution of the Python interpreter and libraries is governed by the Python Software License included with this file, or by other licenses as marked.

Copyright (c) 1998-2000 Thai Open Source Software Center Ltd and Clark Cooper  
Copyright (c) 2001-2022 Expat maintainers

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

#### A. HISTORY OF THE SOFTWARE

=====

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see <https://www.cwi.nl>) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see <https://www.cnri.reston.va.us>) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations, which became Zope Corporation. In 2001, the Python Software Foundation (PSF, see <https://www.python.org/psf/>) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation was a sponsoring member of the PSF.

All

Python releases are Open Source (see <https://opensource.org> for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

| Release        | Derived from | Year      | Owner      | GPL-compatible? (1) |
|----------------|--------------|-----------|------------|---------------------|
| 0.9.0 thru 1.2 |              | 1991-1995 | CWI        | yes                 |
| 1.3 thru 1.5.2 | 1.2          | 1995-1999 | CNRI       | yes                 |
| 1.6            | 1.5.2        | 2000      | CNRI       | no                  |
| 2.0            | 1.6          | 2000      | BeOpen.com | no                  |
| 1.6.1          | 1.6          | 2001      | CNRI       | yes (2)             |
| 2.1            | 2.0+1.6.1    | 2001      | PSF        | no                  |
| 2.0.1          | 2.0+1.6.1    | 2001      | PSF        | yes                 |
| 2.1.1          | 2.1+2.0.1    | 2001      | PSF        | yes                 |
| 2.1.2          | 2.1.1        | 2002      | PSF        | yes                 |
| 2.1.3          | 2.1.2        | 2002      | PSF        | yes                 |
| 2.2 and above  | 2.1.1        | 2001-now  |            |                     |
| PSF            | yes          |           |            |                     |

Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

## B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

Python software and documentation are licensed under the Python Software Foundation License Version 2.

Starting with Python 3.8.6, examples, recipes, and other code in the documentation are dual licensed under the PSF License Version 2 and the Zero-Clause BSD license.

Some software incorporated into Python is under different licenses. The licenses are listed with code falling under that license.

### PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.
2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023 Python Software Foundation; All Rights Reserved" are retained in Python alone or in any derivative version prepared by Licensee.
3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.
4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS

FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

#### BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

-----

#### BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE



WILL NOT  
INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at <http://www.pythonlabs.com/logos.html> may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

#### CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

-----

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the

quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the internet using the following URL: <http://hdl.handle.net/1895.22/1013>".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying,

installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

#### CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

-----

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER

RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

#### ZERO-CLAUSE BSD LICENSE FOR CODE IN THE PYTHON DOCUMENTATION

-----

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

This license applies to the bootstrapper application that is embedded within the installer. It has no impact on the licensing for the rest of the installer or Python itself, as no code covered by this license exists in any other part of the product.

---

## Microsoft Reciprocal License (MS-RL)

This license governs use of the accompanying software. If you use the software, you accept this license. If you do not accept the license, do not use the software.

### 1. Definitions

The terms "reproduce," "reproduction," "derivative works," and "distribution" have the same meaning here as under U.S. copyright law.

A "contribution" is the original software, or any additions or changes to the software.

A "contributor" is any person that distributes its contribution under this license.

"Licensed patents" are a contributor's patent claims that read directly on its contribution.

### 2. Grant of Rights

(A) Copyright Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free copyright license to reproduce its contribution, prepare derivative works of its contribution, and distribute its contribution or any derivative works that you create.

(B) Patent Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free license under its licensed patents to make, have made, use, sell, offer for sale, import, and/or otherwise dispose of its contribution in the software or derivative works of the contribution in the software.

### 3. Conditions and Limitations

(A) Reciprocal Grants- For any file you distribute that contains code from the software (in source code or binary format), you must provide recipients the source code to that file along with a copy of this license, which license will govern that file. You may license other files that are entirely your own work and do not contain code from the software under any terms you choose.

(B) No Trademark License- This license does not grant you rights to use any contributors' name, logo, or trademarks.

(C) If you bring a patent claim against any contributor over patents that you claim are infringed by the software, your patent license from such contributor to the software ends automatically.

(D) If you distribute any portion of the software, you must retain all copyright, patent, trademark, and attribution notices that are present in the software.

(E) If you distribute any portion of the software in source code form, you may do so only under this license by including a complete copy of this license with your distribution. If you distribute any portion of the software in compiled or object code form, you may only do so under a license that complies with this license.

(F) The software is licensed "as-is." You bear the risk of using it. The contributors give no express warranties, guarantees or conditions.

You may have additional consumer rights under your local laws which this license cannot change. To the extent permitted under your local laws, the contributors exclude the implied warranties of merchantability, fitness for a particular purpose and non-infringement.

# 1.164 jaraco.context 5.3.0

## 1.164.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN

THE SOFTWARE.

MIT

## 1.165 libsmi 0.5.0-r4

### 1.165.1 Available under license :

Copyright (c) 1999-2008 Frank Strauss, Technical University of Braunschweig.

This software is copyrighted by Frank Strauss, the Technical University of Braunschweig, and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE

POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

-----  
lib/snprintf.c, lib/snprintf.h:

\* Copyright (c) 1995-1999 Kungliga Tekniska Hgskolan  
\* (Royal Institute of Technology, Stockholm, Sweden).  
\* All rights reserved.  
\*  
\* Redistribution and use in source and binary forms, with or without  
\* modification, are permitted provided that the following conditions  
\* are met:  
\*  
\* 1. Redistributions of source code  
\* must retain the above copyright  
\* notice, this list of conditions and the following disclaimer.  
\*  
\* 2. Redistributions in binary form must reproduce the above copyright  
\* notice, this list of conditions and the following disclaimer in the  
\* documentation and/or other materials provided with the distribution.  
\*  
\* 3. Neither the name of the Institute nor the names of its contributors  
\* may be used to endorse or promote products derived from this software  
\* without specific prior written permission.  
\*  
\* THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS ``AS IS" AND  
\* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
\* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
\* ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE  
\* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL  
\* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS  
\* OR SERVICES;  
\* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
\* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT  
\* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY  
\* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF  
\* SUCH DAMAGE.

-----

tools/shhopt.c, tools/shhopt.h:

Title: shhopt - library for parsing command line options.

Version: 1.1.2

Entered-date: 23MAR97

Description: C-functions for parsing command line options, both  
traditional one-character options, and GNU'ish  
--long-options.

Keywords: programming, library, lib, commandline, options

Author: s.h.huseby@usit.uio.no (Sverre H. Huseby)

Primary-site: <http://www.ifi.uio.no/~sverrehu/pub-unix/>

Alternate-site: [sunsite.unc.edu /pub/Linux/libs  
shhopt-1.1.2.tar.gz](http://sunsite.unc.edu/pub/Linux/libs/shhopt-1.1.2.tar.gz)

Platforms: Requires ANSI C-compiler.

Copying-policy: BeerWare: If you have the time and  
money, send me a bottle  
of your favourite beer. If not, just send me a mail or  
something. Copy and use as you wish; just leave the  
author's name where you find it.

---

# 1.166 strace 6.13-r0

## 1.166.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this  
license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts  
as the successor of the GNU Library Public License, version 2, hence  
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom  
to share and change it. By contrast, the GNU General Public Licenses are  
intended to guarantee your freedom to share and change free software--to  
make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you.

You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary



GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of

it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers  
to a function or a  
table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work

under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public

License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with

the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it

contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may

be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time

a

copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library"

must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on

the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact

that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY



15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Libraries

If you develop a new library,  
and you want it to be of the greatest  
possible use to the public, we recommend making it free software that  
everyone can redistribute and change. You can do so by permitting  
redistribution under these terms (or, alternatively, under the terms of  
the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is  
safest to attach them to the start of each source file to most effectively  
convey the exclusion of warranty; and each file should have at least the  
"copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and an idea of what it does.>

Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it  
under the terms of the GNU Lesser General Public License as published  
by the Free Software Foundation; either version 2.1 of the License, or  
(at your option) any later version.

This library is distributed in the hope that it will be useful, but  
WITHOUT

ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in

the library 'Frob' (a library for tweaking knobs) written

by James Random Hacker.

< signature of Ty Coon > , 1 April 1990

Ty Coon, President of Vice

That's all there is to it!

Copyright (c) 2011-2025 The strace developers.

All rights reserved.

strace test suite is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

strace test suite is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See GPL-2.0-or-later for more details. This is the Debian packaged version of strace. For a complete list of changes from the upstream version please see the changelog.

The upstream sources can be found at <https://github.com/strace/strace/>

This is the copyright as found in the upstream sources:

Copyright (c) 1991, 1992 Paul Kranenburg <pk@cs.few.eur.nl>

Copyright (c) 1993 Branko Lankester <branko@hacktic.nl>

Copyright (c) 1993 Ulrich Pegelow <pegelow@moorea.uni-muenster.de>

Copyright (c) 1995, 1996 Michael Elizabeth Chastain <mec@duracef.shout.net>

Copyright (c) 1993, 1994, 1995, 1996 Rick Sladkey <jrs@world.std.com>

Copyright (c) 1998-2001 Wichert Akkerman <wakkerma@deephackmode.org>

Copyright (c) 2001-2025 The strace developers.

All rights reserved.

strace is free software; you can redistribute it and/or modify it  
under the terms of the GNU Lesser General Public License as published  
by the Free Software Foundation; either version 2.1 of the License,  
or (at your option) any later version.

strace is distributed in the hope

that it will be useful, but

WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY  
or FITNESS FOR A PARTICULAR PURPOSE. See LGPL-2.1-or-later for more details.

strace test suite is provided under the terms of the GNU General Public License  
version 2 or later, see tests/COPYING for more details.

Copyright (c) 1991, 1992 Paul Kranenburg <pk@cs.few.eur.nl>

Copyright (c) 1993 Branko Lankester <branko@hacktic.nl>

Copyright (c) 1993 Ulrich Pegelow <pegelow@moorea.uni-muenster.de>

Copyright (c) 1995, 1996 Michael Elizabeth Chastain <mec@duracef.shout.net>

Copyright (c) 1993, 1994, 1995, 1996 Rick Sladkey <jrs@world.std.com>

Copyright (c) 1998-2001 Wichert Akkerman <wakkerma@deephackmode.org>

Copyright (c) 2001-2025 The strace developers.

All rights reserved.

strace is free software; you can redistribute it and/or modify it  
under the terms of the GNU Lesser General Public License as published  
by the Free Software Foundation; either version 2.1 of the License,  
or (at your option) any later version.

strace is distributed in the hope that it will be useful, but

WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY  
or FITNESS FOR A PARTICULAR PURPOSE. See LGPL-2.1-or-later for more details.

strace test suite is provided under the terms of the GNU General Public License  
version

2 or later, see tests/COPYING for more details.

The `bundled/linux' directory contains some of Linux kernel UAPI header files  
copied verbatim from the Linux kernel for compatibility purposes.

These header files are provided by the Linux kernel under the terms of the  
GNU General Public License version 2 only with an explicit syscall exception,  
see bundled/linux/COPYING for more details.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 , USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the

freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for

each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software

patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that

in whole or  
in part contains or is derived from the Program or  
any part thereof, to be licensed as a whole at no charge to all  
third parties under the terms of this License.

c) If the modified program normally reads commands interactively  
when run, you must cause it, when started running for such  
interactive use in the most ordinary way, to print or display  
an announcement including an appropriate copyright notice and  
a notice that there is no warranty (or else, saying that you  
provide a warranty) and that users may redistribute the program  
under these conditions, and telling the user how to view a copy  
of this License. (Exception: if the Program itself is interactive  
but does not normally print such an announcement, your work based  
on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If  
identifiable sections of that work are not derived from the Program,  
and can be reasonably considered independent and separate works in  
themselves, then this License, and its terms, do not apply to those  
sections when you distribute them as separate works. But when you  
distribute the same sections as part of a whole which is a work  
based on the Program, the distribution of the whole must be on the  
terms of this License, whose permissions for other licensees extend  
to the entire whole, and thus to each and every part regardless of  
who wrote it.

Thus, it is not the intent of this section to claim rights or contest  
your rights to work written entirely by you; rather, the intent is  
to exercise the right to control the distribution of derivative or  
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program  
with the Program (or with a work based on the Program) on a volume  
of a storage or distribution medium does not bring the other work  
under the scope of this  
License.

3. You may copy and distribute the Program (or a work based on it,  
under Section 2) in object code or executable form under the terms of  
Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable  
source code, which must be distributed under the terms of Sections  
1 and 2 above on a medium customarily used for software interchange;  
or,

b) Accompany it with a written offer, valid for at least three

years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed

only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to

copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot

distribute so as to satisfy simultaneously your obligations under this

License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.



9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. END OF TERMS AND CONDITIONS

## How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and an idea of what it does.>

Copyright (C) <yyyy> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 , USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or

your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

< signature of Ty Coon > , 1 April 1989 Ty Coon, President of Vice This  
General Public License does not permit incorporating  
your program into  
proprietary programs. If your program is a subroutine library, you may  
consider it more useful to permit linking proprietary applications with  
the library. If this is what you want to do, use the GNU Lesser General  
Public License instead of this License.

The `linux' directory contains some of Linux kernel UAPI header files  
copied verbatim from the Linux kernel for compatibility purposes.

These header files are not a part of the strace project,  
they are provided by the Linux Kernel under:

SPDX-License-Identifier: GPL-2.0 WITH Linux-syscall-note

Being under the terms of the GNU General Public License version 2 only,  
according with:

GPL-2.0

With an explicit syscall exception, as stated at:

Linux-syscall-note

# 1.167 libnettle6 3.10.1-r0

## 1.167.1 Available under license :

GNU GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your  
freedom to share and change it. By contrast, the GNU General Public  
License is intended to guarantee your freedom to share and change free  
software--to make sure the software is free for all its users. This

General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a

notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such

an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues),

conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any



later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively

convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into

proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

## GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>  
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

### 0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

### 1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

## 2. Conveying

### Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

- a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or
- b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

## 3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

- a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the object code with a copy of the GNU GPL and this license document.

## 4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

- a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are

covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under

terms of your  
choice, if you do both of the following:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
- b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

#### 6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

#### GNU GENERAL PUBLIC LICENSE Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>  
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast,

the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand

ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS

### 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2)



tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

#### 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The

"System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that

same work.

## 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered

by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

## 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

## 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you

receive it, in any medium, provided that you conspicuously and appropriately  
publish on each copy an appropriate copyright notice;  
keep intact all notices stating that this License and any  
non-permissive terms added in accord with section 7 apply to the code;  
keep intact all notices of the absence of any warranty; and give all  
recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey,  
and you may offer support or warranty protection for a fee.

#### 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to  
produce it from the Program, in the form of source code under the  
terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified  
it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is  
released under this License and any conditions added under section  
7. This requirement modifies the requirement in section  
4 to  
"keep intact all notices".
- c) You must license the entire work, as a whole, under this  
License to anyone who comes into possession of a copy. This  
License will therefore apply, along with any applicable section 7  
additional terms, to the whole of the work, and all its parts,  
regardless of how they are packaged. This License gives no  
permission to license the work in any other way, but it does not  
invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display  
Appropriate Legal Notices; however, if the Program has interactive  
interfaces that do not display Appropriate Legal Notices, your  
work need not make them do so.

A compilation of a covered work with other separate and independent  
works, which are not by their nature extensions of the covered work,  
and which are not combined with it such as to form a larger program,  
in or on a volume of a storage or distribution  
medium, is called an  
"aggregate" if the compilation and its resulting copyright are not  
used to limit the access or legal rights of the compilation's users  
beyond what the individual works permit. Inclusion of a covered work  
in an aggregate does not cause this License to apply to the other

parts of the aggregate.

## 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided

you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and

adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

## 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some

trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on

those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

## 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that

copyright

holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

#### 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

#### 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.



## 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify

or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

## 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

## 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

#### 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

#### 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

#### 16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),

EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17.

Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>

This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<https://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <<https://www.gnu.org/philosophy/why-not-lgpl.html>>.

## 1.168 distlib 0.3.9

### 1.168.1 Available under license :

#### A. HISTORY OF THE SOFTWARE

=====

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see <http://www.cwi.nl>) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see <http://www.cnri.reston.va.us>) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations (now Zope Corporation, see <http://www.zope.com>). In 2001, the Python Software Foundation (PSF, see <http://www.python.org/psf/>) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation is a sponsoring member of the PSF.

All Python releases are Open Source (see <http://www.opensource.org> for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

| Release        | Derived from | Year      | Owner      | GPL-compatible? (1) |
|----------------|--------------|-----------|------------|---------------------|
| 0.9.0 thru 1.2 |              | 1991-1995 | CWI        | yes                 |
| 1.3 thru 1.5.2 | 1.2          | 1995-1999 | CNRI       | yes                 |
| 1.6            | 1.5.2        | 2000      | CNRI       | no                  |
| 2.0            | 1.6          | 2000      | BeOpen.com | no                  |
| 1.6.1          | 1.6          | 2001      | CNRI       | yes (2)             |
| 2.1            | 2.0+1.6.1    | 2001      | PSF        | no                  |
| 2.0.1          | 2.0+1.6.1    | 2001      | PSF        | yes                 |
| 2.1.1          | 2.1+2.0.1    | 2001      | PSF        | yes                 |
| 2.2            | 2.1.1        | 2001      | PSF        | yes                 |
| 2.1.2          | 2.1.1        | 2002      | PSF        | yes                 |
| 2.1.3          |              |           |            |                     |
| 2.1.2          | 2002         | PSF       | yes        |                     |
| 2.2.1          | 2.2          | 2002      | PSF        | yes                 |
| 2.2.2          | 2.2.1        | 2002      | PSF        | yes                 |
| 2.2.3          | 2.2.2        | 2003      | PSF        | yes                 |
| 2.3            | 2.2.2        | 2002-2003 | PSF        | yes                 |
| 2.3.1          | 2.3          | 2002-2003 | PSF        | yes                 |
| 2.3.2          | 2.3.1        | 2002-2003 | PSF        | yes                 |
| 2.3.3          | 2.3.2        | 2002-2003 | PSF        | yes                 |
| 2.3.4          | 2.3.3        | 2004      | PSF        | yes                 |
| 2.3.5          | 2.3.4        | 2005      | PSF        | yes                 |
| 2.4            | 2.3          | 2004      | PSF        | yes                 |
| 2.4.1          | 2.4          | 2005      | PSF        | yes                 |
| 2.4.2          | 2.4.1        | 2005      | PSF        | yes                 |
| 2.4.3          | 2.4.2        | 2006      | PSF        | yes                 |
| 2.4.4          | 2.4.3        | 2006      | PSF        | yes                 |
| 2.5            | 2.4          | 2006      | PSF        | yes                 |
| 2.5.1          | 2.5          | 2007      | PSF        | yes                 |
| 2.5.2          | 2.5.1        | 2008      | PSF        | yes                 |
| 2.5.3          | 2.5.2        | 2008      | PSF        | yes                 |
| 2.6            | 2.5          | 2008      | PSF        | yes                 |
| 2.6.1          | 2.6          | 2008      | PSF        | yes                 |
| 2.6.2          | 2.6.1        | 2009      | PSF        | yes                 |
| 2.6.3          | 2.6.2        | 2009      | PSF        | yes                 |
| 2.6.4          | 2.6.3        | 2009      | PSF        | yes                 |
| 2.6.5          | 2.6.4        | 2010      | PSF        | yes                 |
| 3.0            | 2.6          | 2008      | PSF        | yes                 |
| 3.0.1          | 3.0          | 2009      | PSF        | yes                 |
| 3.1            | 3.0.1        | 2009      | PSF        | yes                 |

|       |     |      |     |     |
|-------|-----|------|-----|-----|
| 3.1.1 | 3.1 | 2009 | PSF | yes |
| 3.1.2 | 3.1 | 2010 | PSF | yes |
| 3.2   | 3.1 | 2010 | PSF | yes |

Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

## B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

### PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010 Python Software Foundation; All Rights Reserved" are retained in Python alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

#### BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

-----

#### BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.



3.

BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at <http://www.pythonlabs.com/logos.html> may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

#### CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

-----

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's

License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the Internet using the following URL: <http://hdl.handle.net/1895.22/1013>".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6.

This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This

License Agreement does not grant permission  
to use CNRI trademarks or  
trade name in a trademark sense to endorse or promote products or  
services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying,  
installing or otherwise using Python 1.6.1, Licensee agrees to be  
bound by the terms and conditions of this License Agreement.

ACCEPT

#### CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

-----  
Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam,  
The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its  
documentation for any purpose and without fee is hereby granted,  
provided that the above copyright notice appear in all copies and that  
both that copyright notice and this permission notice appear in  
supporting documentation, and that the name of Stichting Mathematisch  
Centrum or CWI not be used in advertising or publicity pertaining to  
distribution of the software  
without specific, written prior  
permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO  
THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND  
FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE  
FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES  
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN  
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT  
OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

=====  
Distutils2 Contributors

=====  
The Distutils2 project was started by Tarek Ziad and is currently  
maintained by ric Araujo. Many people have contributed to the project.

distlib has started off using some of the code from distutil2.

If you're making a patch, please add your name below in alphabetical order,  
and welcome into the Fellowship of the Packaging!

Thanks to:

- Rajiv Abraham
- Ali Afshar
- David Barnett
- Pior Bastida
- Anthony Baxter
- Erik Bray
- C. Titus Brown
- Francisco Martn Brugu
- Nicolas Cadou
- Godefroid Chapelle
- Julien Courteau
- Christophe Combelles
- Jason R. Coombs
- Pierre-Yves David
- Ned Deily
- Konrad DeLong
- Josip Djolonga
- John Edmonds
- Andr Espaze
- Boris Feld
- Andrew Francis
- Hallvard B Furuseth
- Patrice Gauthier
- Yannick Gingras
- Filip Gruszczyski
- Walker Hale IV
- Alexandre Hamelin
- Kelsey Hightower
- Thomas Holmes
- Preston Holmes
- Christian Hudon
- Julien Jehannet
- Jeremy Kloth
- Thomas Kluyver
- 
- Amos Latteier
- Mathieu Leduc-Hamel
- Pierre Paul Lefebvre
- Tshepang Lekhonkhobe
- Alain Leufroy
- Janusz Lewandowski
- Martin von Lwis
- Hugo Lopes Tavares
- Guillermo Lpez-Anglada
- Justin Love
- Simon Mathieu
- Carl Meyer

- Alexis Mtaireau
- Julien Miotte
- Zubin Mithra
- Derek McTavish Mounce
- Paul Moore
- Michael Mulich
- Louis Munro
- Gal Pasgrimaud
- George Peristerakis
- Mathieu Perreault
- Guillaume Pratte
- Sean Reifschneider
- Antoine Reversat
- Arc Riley
- C. Anthony Risinger
- Elson Rodriguez
- Luis Rojas
- Erik Rose
- Brian Rosner
- Vinay Sajip
- Victor Stinner
- Alexandre Vassalotti
- Nadeem Vawda

Copyright (C) 2013 by Test User.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Vinay Sajip not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

VINAY SAJIP DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL VINAY SAJIP BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.  
BSD-licensed.

# 1.169 clang 18.1.8-r5

## 1.169.1 Available under license :

```
=====
The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:
=====
```

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,

the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition,

"submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

- (a) You must give any other recipients of the Work or  
Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices  
stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works  
that You distribute, all copyright, patent, trademark, and  
attribution notices from the Source form of the Work,  
excluding those notices that do not pertain to any part of  
the Derivative Works; and
- (d) If the Work  
includes a "NOTICE" text file as part of its  
distribution, then any Derivative Works that You distribute must  
include a readable copy of the attribution notices contained  
within such NOTICE file, excluding those notices that do not  
pertain to any part of the Derivative Works, in at least one  
of the following places: within a NOTICE text file distributed  
as part of the Derivative Works; within the Source form or  
documentation, if provided along with the Derivative Works; or,  
within a display generated by the Derivative Works, if and  
wherever such third-party notices normally appear. The contents  
of the NOTICE file are for informational purposes only and  
do not modify the License. You may add Your own attribution  
notices within Derivative Works that You distribute, alongside  
or as an addendum to the NOTICE text from the Work, provided  
that such additional  
attribution notices cannot be construed  
as modifying the License.

You may add Your own copyright statement to Your modifications and  
may provide additional or different license terms and conditions  
for use, reproduction, or distribution of Your modifications, or  
for any such Derivative Works as a whole, provided Your use,  
reproduction, and distribution of the Work otherwise complies with  
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,  
any Contribution intentionally submitted for inclusion in the Work  
by You to the Licensor shall be under the terms and conditions of  
this License, without any additional terms or conditions.  
Notwithstanding the above, nothing herein shall supersede or modify  
the terms of any separate license agreement you may have executed  
with Licensor regarding such Contributions.



6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

#### ---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

=====  
Software from third parties included in the LLVM Project:  
=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or `LICENSE` file at the top containing the specific license and restrictions which apply to that software, or

2) It will contain specific license and restriction terms at the top of every file.

=====  
Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):  
=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2007-2019 University of Illinois at Urbana-Champaign.  
All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

<http://llvm.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- \* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the

power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition,

"submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

#### ---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

=====  
Software from third parties included in the LLVM Project:  
=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

1) It will be in a separate directory tree with its own `LICENSE.txt` or



`LICENSE` file at the top containing the specific  
license and restrictions  
which apply to that software, or  
2) It will contain specific license and restriction terms at the top of every  
file.

=====

Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):

=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2007-2018 University of Illinois at Urbana-Champaign.  
All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

<http://llvm.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

\* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.

\* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

# 1.170 inflect 7.3.1

## 1.170.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN

THE SOFTWARE.

MIT

# 1.171 msgpack 1.1.0

## 1.171.1 Available under license :

MIT License

Copyright (c) 2018 Cees-Jan Kiewiet

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.172 zmap 4.2.0-r0

## 1.172.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# 1.173 bcc 0.34.0-r0

## 1.173.1 Available under license :

Format: <http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Upstream-Name: bcc

Source: <https://github.com/iovisor/bcc>

Files: \*

Copyright: 2015 PLUMgrid, Inc.

License: Apache-2.0

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You



institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use,

reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# 1.174 freeipmi 18.0.2

## 1.174.1 Available under license :

Copyright (C) 2004-2007 The Regents of the University of California.  
Produced at Lawrence Livermore National Laboratory.  
Written by Albert Chu <chu11@llnl.gov>  
UCRL-CODE-155913

This file is part of Bmc-watchdog, a base management controller (BMC) watchdog timer management tool. For details, see <http://www.llnl.gov/linux/>.

Bmc-Watchdog is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Bmc-Watchdog is distributed in the hope that it will be useful, but WITHOUT

ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with Bmc-Watchdog; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

This notice is required to be provided under our contract with the U.S. Department of Energy (DOE). This work was produced at the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-ENG-48 with the DOE.

Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately-owned rights.

Also, reference herein to any specific commercial products, process, or services by trade name, trademark, manufacturer or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

The precise terms and conditions for copying, distribution and modification are specified in the file "COPYING".  
Copyright (C) 2006-2007 The Regents of the University of California.  
Produced at Lawrence Livermore National Laboratory.  
Written by Albert Chu <chu11@llnl.gov>  
UCRL-CODE-222073

This file is part of Ipmimonitoring, an IPMI sensor monitoring library. For details, see <http://www.llnl.gov/linux/>.

Ipmimonitoring is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Ipmimonitoring is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of

MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with Ipmimonitoring; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

This notice is required to be provided under our contract with the U.S. Department of Energy (DOE). This work was produced at the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-ENG-48 with the DOE.

Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately-owned rights.

Also, reference herein to any specific commercial products, process, or services by trade name, trademark, manufacturer or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the

University of California, and shall not be used for advertising or product endorsement purposes.

The precise terms and conditions for copying, distribution and modification are specified in the file "COPYING".  
Copyright (C) 2007 The Regents of the University of California.  
Produced at Lawrence Livermore National Laboratory.  
Written by Albert Chu <chu11@llnl.gov>  
UCRL-CODE-228523

This file is part of Ipmidetector, tools and libraries for detecting IPMI nodes in a cluster. For details, see <http://www.llnl.gov/linux/>.

Ipmidetector is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Ipmidetector is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or

FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with Ipmitect; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

This notice is required to be provided under our contract with the U.S. Department of Energy (DOE). This work was produced at the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-ENG-48 with the DOE.

Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately-owned rights.

Also, reference herein to any specific commercial products, process, or services by trade name, trademark, manufacturer or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

The precise terms and conditions for copying, distribution and modification are specified in the file "COPYING".  
Copyright (C) 2007 The Regents of the University of California.  
Produced at Lawrence Livermore National Laboratory.  
Written by Albert Chu <chu11@llnl.gov>  
UCRL-CODE-232183

This file is part of Ipmi-fru, a tool used for retrieving motherboard field replaceable unit (FRU) information. For details, see <http://www.llnl.gov/linux/>.

Ipmi-fru is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Ipmi-fru is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or

FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with Ipmitool; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

This notice is required to be provided under our contract with the U.S. Department of Energy (DOE). This work was produced at the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-ENG-48 with the DOE.

Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately-owned rights.

Also, reference herein to any specific commercial products, process, or services by trade name, trademark, manufacturer or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

The precise terms and conditions for copying, distribution and modification are specified in the file "COPYING".  
Copyright (C) 2003-2007 The Regents of the University of California.  
Produced at Lawrence Livermore National Laboratory.  
Written by Albert Chu <chu11@llnl.gov>  
UCRL-CODE-155448

This file is part of Ipmitool, tools for ping IPMI and RMCP compliant remote systems. For details, see <http://www.llnl.gov/linux/>.

Ipmitool is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Ipmitool is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License

for more details.

You should have received a copy of the GNU General Public License along with Ipmping; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

This notice is required to be provided under our contract with the U.S. Department of Energy (DOE). This work was produced at the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-ENG-48 with the DOE.

Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately-owned rights.

Also, reference herein to any specific commercial products, process, or services by trade name, trademark, manufacturer or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

The precise terms and conditions for copying, distribution and modification are specified in the file "COPYING".  
Copyright (C) 2007 The Regents of the University of California.  
Produced at Lawrence Livermore National Laboratory.  
Written by Albert Chu <chu11@llnl.gov>  
UCRL-CODE-227589

This file is part of pstdout, a library used to launch and manage the standard output of multiple threads. For details, see <http://www.llnl.gov/linux/>.

Pstdout is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Pstdout is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License



for more details.

You should have received a copy of the GNU General Public License along with Pstdout; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

This notice is required to be provided under our contract with the U.S. Department of Energy (DOE). This work was produced at the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-ENG-48 with the DOE.

Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately-owned rights.

Also, reference herein to any specific commercial products, process, or services by trade name, trademark, manufacturer or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

The precise terms and conditions for copying, distribution and modification are specified in the file "COPYING".  
Copyright (C) 2003-2007 The Regents of the University of California.  
Produced at Lawrence Livermore National Laboratory.  
Written by Albert Chu <chu11@llnl.gov>  
UCRL-CODE-155698

This file is part of Ipmipower, a remote power control utility. For details, see <http://www.llnl.gov/linux/>.

Ipmipower is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Ipmipower is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with Ipmipower; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

This notice is required to be provided under our contract with the U.S. Department of Energy (DOE). This work was produced at the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-ENG-48 with the DOE.

Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately-owned rights.

Also, reference herein to any specific commercial products, process, or services by trade name, trademark, manufacturer or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

The precise terms and conditions for copying, distribution and modification are specified in the file "COPYING".  
Copyright (C) 2006-2007 The Regents of the University of California.  
Produced at Lawrence Livermore National Laboratory.  
Written by Albert Chu <chu11@llnl.gov>  
UCRL-CODE-221226

This file is part of Ipmiconsole, a set of IPMI 2.0 SOL libraries and utilities. For details, see <http://www.llnl.gov/linux/>.

Ipmiconsole is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Ipmiconsole is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with Ipmiconsole; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

This notice is required to be provided under our contract with the U.S. Department of Energy (DOE). This work was produced at the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-ENG-48 with the DOE.

Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately-owned rights.

Also, reference herein to any specific commercial products, process, or services by trade name, trademark, manufacturer or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

The precise terms and conditions for copying, distribution and modification are specified in the file "COPYING". The file libfreeipmi/driver/freeipmi\_bmc\_intf.h is from ipmitool (<http://sourceforge.net/projects/ipmitool>) and comes with the following license.

----

Copyright (c) 2003 Sun Microsystems, Inc. All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistribution of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistribution in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Sun Microsystems, Inc. or the names of

contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided "AS IS," without a warranty of any kind.

ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING

ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED.

SUN MICROSYSTEMS, INC. ("SUN") AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

----

#### GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have

certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1)

assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand

ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS

### 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

#### 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The

"System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

## 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you

with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

### 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

### 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

### 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:



- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

## 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years

and valid for as

long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user

actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

## 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions

apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does

not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

## 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

## 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or

modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

#### 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

#### 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this

License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that

contain the covered work, unless you entered into that arrangement,  
or  
that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting  
any implied license or other defenses to infringement that may  
otherwise be available to you under applicable patent law.

#### 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or  
otherwise) that contradict the conditions of this License, they do not  
excuse you from the conditions of this License. If you cannot convey a  
covered work so as to satisfy simultaneously your obligations under this  
License and any other pertinent obligations, then as a consequence you may  
not convey it at all. For example, if you agree to terms that obligate you  
to collect a royalty for further conveying from those to whom you convey  
the Program, the only way you could satisfy both those terms and this  
License would be to refrain entirely from conveying the Program.

#### 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have  
permission to link or combine any covered work with a work licensed  
under version 3 of the GNU Affero General Public License into a single  
combined work, and to convey the resulting work. The terms of this  
License will continue to apply to the part which is the covered work,  
but the special requirements of the GNU Affero General Public License,  
section 13, concerning interaction through a network will apply to the  
combination as such.

#### 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of  
the GNU General Public License from time to time. Such new versions will  
be similar in spirit to the present version, but may differ in detail to  
address new problems or concerns.

Each version is given a distinguishing version number. If the  
Program specifies that a certain numbered version of the GNU General  
Public License "or any later version" applies to it, you  
have the  
option of following the terms and conditions either of that numbered  
version or of any later version published by the Free Software  
Foundation. If the Program does not specify a version number of the  
GNU General Public License, you may choose any version ever published  
by the Free Software Foundation.



If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

#### 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

#### 16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### 17.

Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>

This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<http://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you

may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read [<http://www.gnu.org/philosophy/why-not-lgpl.html>](http://www.gnu.org/philosophy/why-not-lgpl.html).

.PP

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

.PP

FreeIPMI is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

.PP

This library is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

This work was produced at the Lawrence Livermore National Laboratory (LLNL) under Contract No. DE-AC52-07NA27344 (Contract 44) between the U.S. Department of Energy (DOE) and Lawrence Livermore National Security, LLC (LLNS) for the operation of LLNL.

This work was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor Lawrence Livermore National Security, LLC nor any of their employees, makes any warranty, express or implied, or assumes any liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately-owned rights.

Reference herein to any specific commercial products, process, or services by trade name, trademark, manufacturer or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or Lawrence Livermore National

Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

The precise terms and conditions for copying, distribution, and modification are specified in the file "COPYING".

# 1.175 Idns 1.8.4-r0

## 1.175.1 Available under license :

This software is copyright (c) 2013 by UNINETT Norid AS. No license is granted to other entities.

All rights reserved.

Copyright (c) 2005,2006, NLnetLabs

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of NLnetLabs nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO

EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2011, Xelerance

Author: Christopher Olah <chris@xelerance.com>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\* Neither the name of Xelerance nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2009, Zdenek Vasicek (vasicek AT fit.vutbr.cz)

Karel Slany (slany AT fit.vutbr.cz)

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of the organization nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# 1.176 typing-extensions 4.12.2

## 1.176.1 Available under license :

### A. HISTORY OF THE SOFTWARE

=====

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see <https://www.cwi.nl>) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see <https://www.cnri.reston.va.us>) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations, which became Zope Corporation. In 2001, the Python Software Foundation (PSF, see <https://www.python.org/psf/>) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation was a sponsoring member of the PSF.

All

Python releases are Open Source (see <https://opensource.org> for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

| Release        | Derived<br>from | Year      | Owner      | GPL-<br>compatible? (1) |
|----------------|-----------------|-----------|------------|-------------------------|
| 0.9.0 thru 1.2 |                 | 1991-1995 | CWI        | yes                     |
| 1.3 thru 1.5.2 | 1.2             | 1995-1999 | CNRI       | yes                     |
| 1.6            | 1.5.2           | 2000      | CNRI       | no                      |
| 2.0            | 1.6             | 2000      | BeOpen.com | no                      |
| 1.6.1          | 1.6             | 2001      | CNRI       | yes (2)                 |
| 2.1            | 2.0+1.6.1       | 2001      | PSF        | no                      |
| 2.0.1          | 2.0+1.6.1       | 2001      | PSF        | yes                     |
| 2.1.1          | 2.1+2.0.1       | 2001      | PSF        | yes                     |
| 2.1.2          | 2.1.1           | 2002      | PSF        | yes                     |
| 2.1.3          | 2.1.2           | 2002      | PSF        | yes                     |
| 2.2 and above  | 2.1.1           | 2001-now  |            |                         |
| PSF            | yes             |           |            |                         |

Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

## B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

Python software and documentation are licensed under the Python Software Foundation License Version 2.

Starting with Python 3.8.6, examples, recipes, and other code in the documentation are dual licensed under the PSF License Version 2 and the Zero-Clause BSD license.

Some software incorporated into Python is under different licenses. The licenses are listed with code falling under that license.

## PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright

(c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023 Python Software Foundation; All Rights Reserved" are retained in Python alone or in any derivative version

prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

-----

BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License



Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at <http://www.pythonlabs.com/logos.html> may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

#### CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

-----

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the internet using the following URL: <http://hdl.handle.net/1895.22/1013>".
3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.
4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.
5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.
6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.
7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based

on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

#### CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

-----

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER

RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

#### ZERO-CLAUSE BSD LICENSE FOR CODE IN THE PYTHON DOCUMENTATION

-----

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

## 1.177 pam 1.7.0-r4

### 1.177.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the

source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you

conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in

whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or

collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6.

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the



integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES

PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),  
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands

`show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

```
# Automatically generated by apkbuidl-cpan, template 4
# Contributor: Valery Kartel <valery.kartel@gmail.com>
# Maintainer: Celeste <cielesti@protonmail.com>
maintainer="Celeste <cielesti@protonmail.com>"
pkgname=perl-bsd-resource
pkgver=1.2911
pkgrel=10
#_pkgreal is used by apkbuidl-cpan to find modules at MetaCpan
_pkgreal=BSD-Resource
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="https://metacpan.org/release/BSD-Resource/"
arch="all"
license="Artistic-2.0 OR LGPL-2.0"
depends="perl"
makedepends="perl-dev"
```

```

subpackages="$pkgname-doc"
source="https://cpan.metacpan.org/authors/id/J/JH/JHI/BSD-Resource-$pkgver.tar.gz"
builddir="$srcdir/$_pkgreal-$pkgver"

build() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    PERL_MM_USE_DEFAULT=1 perl -I. Makefile.PL \
        INSTALLDIRS=vendor \
        NO_PACKLIST=1 \
        NO_PERLLOCAL=1
    make
}

check() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    make test
}

package() {
    make DESTDIR="$pkgdir" install
}

sha512sums="
d0032d41c7c0468ed1c6d8f57b885f6cb97a5039d754c8cb60b2067daedaf53bd15fb6561a3d0f828df16dfa5417f663b
8065ba65f4fea16dc9262728b3b6b85
    BSD-Resource-1.2911.tar.gz
"

# Contributor: Maxim Karasev <begs@disroot.org>
# Maintainer: mio <miyopan@e.email>
pkgname=bsd-games
pkgver=3.3
pkgrel=2
pkgdesc="traditional text mode games from BSD"
url="https://bsd-games.sourceforge.io/"
arch="all"
license="BSD-3-Clause"
install="$pkgname.post-install"
makedepends="ncurses-dev coreutils" # configure causes busybox expr error
subpackages="$pkgname-doc"
source="https://sourceforge.net/projects/bsd-games/files/bsd-games-$pkgver.tar.gz
0001-bsd-games-scores-dir.patch
"

options="!check" # no tests

build() {
    # some GNU autoconf options are ignored, but it works
    ./configure \
        --build=$CBUILD \

```

```

--host=$CHOST \
--prefix=/usr \
--sysconfdir=/etc \
--mandir=/usr/share/man \
--scoredir=/usr/share/$pkgname
make
}

package() {
make DESTDIR="$pkgdir" install
install -Dm644 LICENSE "$pkgdir"/usr/share/licenses/bsd-games/LICENSE
}

sha512sums="
aaf36d09d4fe68514a5c279063d7e77a9a9a84c447037cba974b29faa07eb80c2aab2379d0699c196b5c27ffe1b2bb2c6
8d3390143e0e62d5e3d210ccfb61294
    bsd-games-3.3.tar.gz
e1def0d4835a877242c4173fae862e278a7858738e0bacacbb045a42bcabec2aa7c72d000bbd2421ce35fe66cdc1f52c0
577b7e278029ca5ed50f7aab22319f6 0001-bsd-games-scores-dir.patch
"

MIT License

```

Copyright (c) 2023 Leaf Corcoran

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# Maintainer: Natanael Copa <ncopa@alpinelinux.org>

pkgname=libart-lgpl

pkgver=2.3.21

pkgrel=8

pkgdesc="A library for high-performance 2D graphics"

url="https://www.levien.com/libart/"

```
arch="all"
options="!check" # No test suite.
license="GPL-2.0-or-later"
subpackages="$pkgname-dev"
source="https://download.gnome.org/sources/libart_lgpl/2.3/libart_lgpl-$pkgver.tar.bz2"
"
```

```
builddir="$srcdir"/libart_lgpl-$pkgver
```

```
prepare() {
    update_config_sub
    default_prepare
}
```

```
build() {
    ./configure \
        --build=$CBUILD \
        --host=$CHOST \
        --prefix=/usr
    make
}
```

```
package() {
    make DESTDIR="$pkgdir" install
}
```

```
sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020 libart_lgpl-2.3.21.tar.bz2"
mini_sendmail - accept email on behalf of real sendmail
```

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.  
All rights reserved.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions  
are met:

1. Redistributions of source code must retain the above copyright  
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright  
notice, this list of conditions and the following disclaimer in the  
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND  
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE  
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL  
DAMAGES (INCLUDING,

BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS  
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT  
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY  
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF  
SUCH DAMAGE.

# Contributor: Patrycja Rosa <alpine@ptrcnull.me>

# Maintainer: Patrycja Rosa <alpine@ptrcnull.me>

pkgname=bsd-compat-headers

pkgver=0.7.2

pkgrel=6

pkgdesc="BSD compatibility headers (cdefs, queue, tree)"

url="https://gitlab.alpinelinux.org/alpine/aports"

arch="noarch"

license="BSD-2-Clause AND BSD-3-Clause"

source="

cdefs.h

queue.h

tree.h

"

builddir="\$srcdir"

options="!check" # just headers

package() {

mkdir -p "\$pkgdir"

install -Dm644 -t "\$pkgdir"/usr/include/sys \

cdefs.h queue.h tree.h

}

sha512sums="

37c8fc73c7aea7b490f7850927e2bb91d12137c9e59e22c084146d515696dbc7973b5de92f4c987ba080dd2502ba8312  
7006442c3f019b6447a620c0cae73178 cdefs.h

2f0d5e6e4dc3350285cf17009265dddcbe12431c111868eea39bc8cb038ab7c1f2acacbb21735c4e9d4a1fd106a8fc0f86  
11ea33987d4faba37dde5ce6da0750 queue.h

d9ac210d81feb8ad2655bc80fb065d3fe20ae4417b32b4a1711e6738a4870140005c13373b5d1846ef3ce5ae6da45f2da  
cef2092881eded0a2e94f6a07752ef3 tree.h

"

# Contributor: Carlo Landmeter <clandmeter@alpinelinux.org>

# Maintainer: Kevin Daudt <kdaudt@alpinelinux.org>

pkgname=spdx-licenses

pkgver=3.26.0

pkgrel=0

pkgdesc="Various data formats for the SPDX License List"

url="https://spdx.org/"

arch="noarch"

options="!check" # no test suite

license="CC-BY-3.0"

source="license-list-data-\$pkgver.tar.gz::https://github.com/spdx/license-list-data/archive/v\$pkgver.tar.gz"

builddir="\$srcdir/license-list-data-\$pkgver"

```
subpackages="$pkgname-list"
```

```
_types="html json rdfa rdft rdfturtle rdfxml template text"
```

```
for type in $_types; do
    subpackages="$subpackages $pkgname-$type:_subpkg"
done
```

```
package() {
    mkdir -p "$pkgdir"
}
```

```
_subpkg() {
    local type=${subpkgname/$pkgname-/}
    pkgdesc="$pkgdesc ($type)"
    install_if="$pkgname=$pkgver-r$pkgrel"
    mkdir -p "$subpkgdir"/usr/share/spdx
    cp -r "$builddir"/$type "$subpkgdir"/usr/share/spdx/
}
```

```
list() {
    pkgdesc="$pkgdesc (licence list)"
    mkdir -p "$subpkgdir"/usr/share/spdx
    local i; for i in "$builddir"/text/*.txt;
    do
        local license=${i##*/}
        echo ${license%.*} >> "$subpkgdir"/usr/share/spdx/license.lst
    done
}
```

```
sha512sums="
67e618d4642dfe4f366935d9514d6f7941d3711e77c49bc1b340ac572d36015981b8f10c72a9f80f1d83e132ae7e5093c
0155e0c0e786e4777ee16d1cb7b885a  license-list-data-3.26.0.tar.gz
"
```

Original Copyright (c) 1985 by Supoj Sutanthavibul  
Parts Copyright (c) 1989-2012 by Brian V. Smith  
Parts Copyright (c) 1991 by Paul King  
Other Copyrights may be found in various files

Main Xfig copyright notice:

Any party obtaining a copy of these files is granted, free of charge, a full and unrestricted irrevocable, world-wide, paid up, royalty-free, nonexclusive right and license to deal in this software and documentation files (the "Software"), including without limitation the rights to use, copy, modify, merge, publish and/or distribute copies of the Software, and to permit persons who receive copies from any such party to do so, with the only requirement being that this copyright notice remain intact.

No representations are made about the suitability of this software for any purpose. It is provided "as is" without



express or implied warranty.

Parts Copyright (C) 1993 by Alan Richardson (mppa3@uk.ac.sussex.syma)

The text rotation code in w\_rottext.c was written by Alan Richardson.

The above copyright notice holds for this work as well.

Parts Copyright (c) 1994 by Anthony Dekker

The Kohonen neural network code for color optimization was written by Anthony Dekker.

Parts Copyright (c) 1995 by C. Blanc and C. Schlick

The X-Spline code was written Carole Blanc (blanc@labri.u-bordeaux.fr) and Christophe Schlick (schlick@labri.u-bordeaux.fr) starting from an initial implementation done by C. Feuille, S. Grosbois, L. Maziere and L. Minihot as a student practice (Universite Bordeaux, France). For additional information about X-splines, see:

"X-Splines: A Spline Model Designed for the End User" by C. Blanc and C. Schlick, Proceedings of SIGGRAPH'95

<http://dept-info.labri.u-bordeaux.fr/~schlick/DOC/sig1.html>

# Contributor: Dhruvin Gandhi <contact@dhruvin>

# Maintainer: Patrycja Rosa <alpine@ptrcnull.me>

pkgname=py3-license-expression

\_pkgname=license\_expression

pkgver=30.4.1

pkgrel=0

pkgdesc="Library to parse, compare, simplify and normalize license expressions"

url="https://github.com/nexB/license-expression"

arch="noarch"

license="Apache-2.0"

depends="python3 py3-boolean.py"

makedepends="py3-setuptools py3-setuptools\_scm py3-gpep517 py3-wheel"

checkdepends="py3-pytest py3-pytest-xdist"

subpackages="\$pkgname-pyc"

source="https://files.pythonhosted.org/packages/source/l/\_pkgname/\_pkgname-\$pkgver.tar.gz"

builddir="\$srcdir/\$pkgname-\$pkgver"

```
build() {
    gpep517 build-wheel \
        --wheel-dir .dist \
        --output-fd 3 3>&1 >&2
}
```

```
check() {
    python3 -m venv --clear --without-pip --system-site-packages .testenv
    .testenv/bin/python3 -m installer .dist/*.whl
    .testenv/bin/python3 -m pytest
}
```

```
package() {
    python3 -m installer -d "$pkgdir" \
        .dist/*.whl
}
```

```

sha512sums="
114c630f098939757e49187e35b4e54883266a2ee9ca283724992420ef4c860f9606a8342da0ab74d93627c10805f312
fda3f699cbc7c4fbd47a71e220440fb0
license_expression-30.4.1.tar.gz
"
/* $NetBSD: queue.h,v 1.70 2015/11/02 15:21:23 christos Exp $ */

/*
 * Copyright (c) 1991, 1993
 * The Regents of the University of California. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the University nor the names of its contributors
 * may be used to endorse or promote products derived from this software
 * without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
 * BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * @(#)queue.h 8.5 (Berkeley) 8/20/94
 */

#ifndef _SYS_QUEUE_H_
#define _SYS_QUEUE_H_

/*
 * This file defines five types of data structures: singly-linked lists,
 * lists, simple queues, tail queues, and circular queues.
 *
 * A singly-linked list is headed by a single forward pointer. The

```

- \* elements are singly linked for minimum space and pointer manipulation
- \* overhead at the expense of  $O(n)$  removal for arbitrary elements. New
- \* elements can be added to the list after an existing element or at the
- \* head of the list. Elements being removed from the head of the list
- \* should use the explicit macro for this purpose for optimum
- \* efficiency. A singly-linked list may only be traversed in the forward
- \* direction. Singly-linked lists are ideal for applications with large
- \* datasets and few or no removals or for implementing a LIFO queue.
- \*
- \* A list is headed by a single forward pointer (or an array of forward
- \* pointers for a hash table header). The elements are doubly linked
- \* so that an arbitrary element can be removed without a need to
- \* traverse the list. New elements can be added to the list before
- \* or after an existing element or at the head of the list. A list
- \* may only be traversed in the forward direction.
- \*
- \* A simple queue is headed by a pair of pointers, one the head of the
- \* list and the
- other to the tail of the list. The elements are singly
- \* linked to save space, so elements can only be removed from the
- \* head of the list. New elements can be added to the list after
- \* an existing element, at the head of the list, or at the end of the
- \* list. A simple queue may only be traversed in the forward direction.
- \*
- \* A tail queue is headed by a pair of pointers, one to the head of the
- \* list and the other to the tail of the list. The elements are doubly
- \* linked so that an arbitrary element can be removed without a need to
- \* traverse the list. New elements can be added to the list before or
- \* after an existing element, at the head of the list, or at the end of
- \* the list. A tail queue may be traversed in either direction.
- \*
- \* A circle queue is headed by a pair of pointers, one to the head of the
- \* list and the other to the tail of the list. The elements are doubly
- \* linked so that an arbitrary element can be removed without a need to
- \* traverse the list. New elements
- can be added to the list before or after
- \* an existing element, at the head of the list, or at the end of the list.
- \* A circle queue may be traversed in either direction, but has a more
- \* complex end of list detection.
- \*
- \* For details on the use of these macros, see the queue(3) manual page.
- \*/
- /\*
- \* Include the definition of NULL only on NetBSD because sys/null.h
- \* is not available elsewhere. This conditional makes the header
- \* portable and it can simply be dropped verbatim into any system.

```

* The caveat is that on other systems some other header
* must provide NULL before the macros can be used.
*/

#ifdef __NetBSD__
#include <sys/null.h>
#endif

#ifdef QUEUEDEBUG
# if defined(_KERNEL)
#  define QUEUEDEBUG_ABORT(...) panic(__VA_ARGS__)
# else
#  include <err.h>
#  define QUEUEDEBUG_ABORT(...) err(1, __VA_ARGS__)
# endif
#endif

/*
* Singly-linked List definitions.
*/

#define SLIST_HEAD(name, type) \
struct name { \
    struct type *slh_first; /* first \
    element */ \
}

#define SLIST_HEAD_INITIALIZER(head) \
{ NULL }

#define SLIST_ENTRY(type) \
struct { \
    struct type *sle_next; /* next element */ \
}

/*
* Singly-linked List access methods.
*/

#define SLIST_FIRST(head) ((head)->slh_first)
#define SLIST_END(head) NULL
#define SLIST_EMPTY(head) ((head)->slh_first == NULL)
#define SLIST_NEXT(elm, field) ((elm)->field.sle_next)

#define SLIST_FOREACH(var, head, field) \
for((var) = (head)->slh_first; \
    (var) != SLIST_END(head); \
    (var) = (var)->field.sle_next)

#define SLIST_FOREACH_SAFE(var, head, field, tvar) \
for ((var) = SLIST_FIRST((head)); \

```

```

    (var) != SLIST_END(head) &&  \
    ((tvar) = SLIST_NEXT((var), field), 1);  \
    (var) = (tvar))

/*
 * Singly-linked List functions.
 */
#define SLIST_INIT(head) do {  \
    (head)->slh_first = SLIST_END(head);  \
} while (/*CONSTCOND*/0)

#define SLIST_INSERT_AFTER(slistelm, elm, field) do {  \
    (elm)->field.sle_next = (slistelm)->field.sle_next;  \
    (slistelm)->field.sle_next  \
    = (elm);  \
} while (/*CONSTCOND*/0)

#define SLIST_INSERT_HEAD(head, elm, field) do {  \
    (elm)->field.sle_next = (head)->slh_first;  \
    (head)->slh_first = (elm);  \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE_AFTER(slistelm, field) do {  \
    (slistelm)->field.sle_next =  \
        SLIST_NEXT(SLIST_NEXT((slistelm), field), field);  \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE_HEAD(head, field) do {  \
    (head)->slh_first = (head)->slh_first->field.sle_next;  \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE(head, elm, type, field) do {  \
    if ((head)->slh_first == (elm)) {  \
        SLIST_REMOVE_HEAD((head), field);  \
    }  \
    else {  \
        struct type *curelm = (head)->slh_first;  \
        while(curelm->field.sle_next != (elm))  \
            curelm = curelm->field.sle_next;  \
        curelm->field.sle_next =  \
            curelm->field.sle_next->field.sle_next;  \
    }  \
} while (/*CONSTCOND*/0)

/*
 * List definitions.

```

```

*/
#define LIST_HEAD(name, type) \
struct name { \
    struct type *lh_first; /* first element */ \
}

#define LIST_HEAD_INITIALIZER(head) \
{ NULL }

#define LIST_ENTRY(type) \
struct { \
    struct type *le_next; /* next element */ \
    struct type **le_prev; /* address of previous next element */ \
}

/*
 * List access methods.
 */
#define LIST_FIRST(head) ((head)->lh_first)
#define LIST_END(head) NULL
#define LIST_EMPTY(head) ((head)->lh_first == LIST_END(head))
#define LIST_NEXT(elm, field) ((elm)->field.le_next)

#define LIST_FOREACH(var, head, field) \
for ((var) = ((head)->lh_first); \
     (var) != LIST_END(head); \
     (var) = ((var)->field.le_next))

#define LIST_FOREACH_SAFE(var, head, field, tvar) \
for ((var) = LIST_FIRST((head)); \
     (var) != LIST_END(head) && \
     ((tvar) = LIST_NEXT((var), field), 1); \
     (var) = (tvar))

#define LIST_MOVE(head1, head2) do { \
    LIST_INIT((head2)); \
    if (!LIST_EMPTY((head1))) \
    { \
        (head2)->lh_first = (head1)->lh_first; \
        LIST_INIT((head1)); \
    } \
} while (/*CONSTCOND*/0)

/*
 * List functions.
 */
#ifdef QUEUEDEBUG
#define QUEUEDEBUG_LIST_INSERT_HEAD(head, elm, field) \

```

```

if ((head)->lh_first && \
    (head)->lh_first->field.le_prev != &(head)->lh_first) \
    QUEUEDEBUG_ABORT("LIST_INSERT_HEAD %p %s:%d", (head), \
        __FILE__, __LINE__);
#define QUEUEDEBUG_LIST_OP(elm, field) \
if ((elm)->field.le_next && \
    (elm)->field.le_next->field.le_prev != \
    &(elm)->field.le_next) \
    QUEUEDEBUG_ABORT("LIST_* forw %p %s:%d", (elm), \
        __FILE__, __LINE__); \
if (*(elm)->field.le_prev != (elm)) \
    QUEUEDEBUG_ABORT("LIST_* back %p %s:%d", (elm), \
        __FILE__, __LINE__);
#define QUEUEDEBUG_LIST_POSTREMOVE(elm, field) \
(elm)->field.le_next = (void *)1L; \
(elm)->field.le_prev = (void *)1L;
#else
#define QUEUEDEBUG_LIST_INSERT_HEAD(head, elm, field)
#define QUEUEDEBUG_LIST_OP(elm,
    field)
#define QUEUEDEBUG_LIST_POSTREMOVE(elm, field)
#endif

#define LIST_INIT(head) do { \
    (head)->lh_first = LIST_END(head); \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_AFTER(listelm, elm, field) do { \
    QUEUEDEBUG_LIST_OP((listelm), field) \
    if (((elm)->field.le_next = (listelm)->field.le_next) != \
        LIST_END(head)) \
        (listelm)->field.le_next->field.le_prev = \
            &(elm)->field.le_next; \
    (listelm)->field.le_next = (elm); \
    (elm)->field.le_prev = &(listelm)->field.le_next; \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_BEFORE(listelm, elm, field) do { \
    QUEUEDEBUG_LIST_OP((listelm), field) \
    (elm)->field.le_prev = (listelm)->field.le_prev; \
    (elm)->field.le_next = (listelm); \
    *(listelm)->field.le_prev = (elm); \
    (listelm)->field.le_prev = &(elm)->field.le_next; \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_HEAD(head, elm, field) do { \
    QUEUEDEBUG_LIST_INSERT_HEAD((head), (elm), field) \
    if (((elm)->field.le_next

```

```

= (head)->lh_first) != LIST_END(head))\
(head)->lh_first->field.le_prev = &(elm)->field.le_next;\
(head)->lh_first = (elm); \
(elm)->field.le_prev = &(head)->lh_first; \
} while (/*CONSTCOND*/0)

#define LIST_REMOVE(elm, field) do { \
    QUEUEDEBUG_LIST_OP((elm), field) \
    if ((elm)->field.le_next != NULL) \
        (elm)->field.le_next->field.le_prev = \
            (elm)->field.le_prev; \
    *(elm)->field.le_prev = (elm)->field.le_next; \
    QUEUEDEBUG_LIST_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

#define LIST_REPLACE(elm, elm2, field) do { \
    if (((elm2)->field.le_next = (elm)->field.le_next) != NULL) \
        (elm2)->field.le_next->field.le_prev = \
            &(elm2)->field.le_next; \
    (elm2)->field.le_prev = (elm)->field.le_prev; \
    *(elm2)->field.le_prev = (elm2); \
    QUEUEDEBUG_LIST_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

/*
 * Simple queue definitions.
 */
#define SIMPLEQ_HEAD(name, type) \
struct name { \
    struct \
    type *sqh_first; /* first element */ \
    struct type **sqh_last; /* addr of last next element */ \
}

#define SIMPLEQ_HEAD_INITIALIZER(head) \
{ NULL, &(head).sqh_first }

#define SIMPLEQ_ENTRY(type) \
struct { \
    struct type *sqe_next; /* next element */ \
}

/*
 * Simple queue access methods.
 */
#define SIMPLEQ_FIRST(head) ((head)->sqh_first)
#define SIMPLEQ_END(head) NULL
#define SIMPLEQ_EMPTY(head) ((head)->sqh_first == SIMPLEQ_END(head))

```



```

#define SIMPLEQ_NEXT(elm, field) ((elm)->field.sqe_next)

#define SIMPLEQ_FOREACH(var, head, field) \
for ((var) = ((head)->sqh_first); \
     (var) != SIMPLEQ_END(head); \
     (var) = ((var)->field.sqe_next))

#define SIMPLEQ_FOREACH_SAFE(var, head, field, next) \
for ((var) = ((head)->sqh_first); \
     (var) != SIMPLEQ_END(head) && \
     ((next = ((var)->field.sqe_next)), 1); \
     (var) = (next))

/*
 * Simple queue functions.
 */
#define SIMPLEQ_INIT(head) do { \
    (head)->sqh_first = NULL; \
    (head)->sqh_last \
    = &(head)->sqh_first; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_HEAD(head, elm, field) do { \
    if (((elm)->field.sqe_next = (head)->sqh_first) == NULL) \
        (head)->sqh_last = &(elm)->field.sqe_next; \
    (head)->sqh_first = (elm); \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_TAIL(head, elm, field) do { \
    (elm)->field.sqe_next = NULL; \
    *(head)->sqh_last = (elm); \
    (head)->sqh_last = &(elm)->field.sqe_next; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_AFTER(head, listelm, elm, field) do { \
    if (((elm)->field.sqe_next = (listelm)->field.sqe_next) == NULL) \
        (head)->sqh_last = &(elm)->field.sqe_next; \
    (listelm)->field.sqe_next = (elm); \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE_HEAD(head, field) do { \
    if (((head)->sqh_first = (head)->sqh_first->field.sqe_next) == NULL) \
        (head)->sqh_last = &(head)->sqh_first; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE_AFTER(head, elm, field) do { \
    if

```

```

(((elm)->field.sqe_next = (elm)->field.sqe_next->field.sqe_next) \
 == NULL)    \
(head)->sqh_last = &(elm)->field.sqe_next; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE(head, elm, type, field) do { \
if ((head)->sqh_first == (elm)) { \
SIMPLEQ_REMOVE_HEAD((head), field); \
} else { \
struct type *curelm = (head)->sqh_first; \
while (curelm->field.sqe_next != (elm)) \
curelm = curelm->field.sqe_next; \
if ((curelm->field.sqe_next = \
curelm->field.sqe_next->field.sqe_next) == NULL) \
(head)->sqh_last = &(curelm)->field.sqe_next; \
} \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_CONCAT(head1, head2) do { \
if (!SIMPLEQ_EMPTY((head2))) { \
*(head1)->sqh_last = (head2)->sqh_first; \
(head1)->sqh_last = (head2)->sqh_last; \
SIMPLEQ_INIT((head2)); \
} \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_LAST(head, type, field) \
(SIMPLEQ_EMPTY((head)) ? \
NULL : \
((struct type \
*)(void *) \
((char *)((head)->sqh_last) - offsetof(struct type, field))))

/*
 * Tail queue definitions.
 */
#define _TAILQ_HEAD(name, type, qual) \
struct name { \
qual type *tqh_first; /* first element */ \
qual type *qual *tqh_last; /* addr of last next element */ \
}

#define TAILQ_HEAD(name, type) _TAILQ_HEAD(name, struct type,)

#define TAILQ_HEAD_INITIALIZER(head) \
{ TAILQ_END(head), &(head).tqh_first }

#define _TAILQ_ENTRY(type, qual) \
struct { \

```

```

qual type *tqe_next; /* next element */ \
qual type *qual *tqe_prev; /* address of previous next element */\
}
#define TAILQ_ENTRY(type) _TAILQ_ENTRY(struct type,)

/*
 * Tail queue access methods.
 */
#define TAILQ_FIRST(head) ((head)->tqh_first)
#define TAILQ_END(head) (NULL)
#define TAILQ_NEXT(elm, field) ((elm)->field.tqe_next)
#define TAILQ_LAST(head, headname) \
    (((struct headname *) (void *) ((head)->tqh_last))->tqh_last)
#define TAILQ_PREV(elm, headname, field) \
    (((struct headname *) (void *) ((elm)->field.tqe_prev))->tqh_last)
#define TAILQ_EMPTY(head) (TAILQ_FIRST(head) == TAILQ_END(head))

#define TAILQ_FOREACH(var, head, field) \
    for ((var) = ((head)->tqh_first); \
         (var) != TAILQ_END(head); \
         (var) = ((var)->field.tqe_next))

#define TAILQ_FOREACH_SAFE(var, head, field, next) \
    for ((var) = ((head)->tqh_first); \
         (var) != TAILQ_END(head) && \
         ((next) = TAILQ_NEXT(var, field), 1); (var) = (next))

#define TAILQ_FOREACH_REVERSE(var, head, headname, field) \
    for ((var) = TAILQ_LAST((head), headname); \
         (var) != TAILQ_END(head); \
         (var) = TAILQ_PREV((var), headname, field))

#define TAILQ_FOREACH_REVERSE_SAFE(var, head, headname, field, prev) \
    for ((var) = TAILQ_LAST((head), headname); \
         (var) != TAILQ_END(head) && \
         ((prev) = TAILQ_PREV((var), headname, field), 1); (var) = (prev))

/*
 * Tail queue functions.
 */
#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_TAILQ_INSERT_HEAD(head, \
    elm, field) \
    if ((head)->tqh_first && \
        (head)->tqh_first->field.tqe_prev != &(head)->tqh_first) \
        QUEUEDEBUG_ABORT("TAILQ_INSERT_HEAD %p %s:%d", (head), \

```

```

    __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_INSERT_TAIL(head, elm, field) \
    if (*(head)->tqh_last != NULL) \
        QUEUEDEBUG_ABORT("TAILQ_INSERT_TAIL %p %s:%d", (head), \
            __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_OP(elm, field) \
    if ((elm)->field.tqe_next && \
        (elm)->field.tqe_next->field.tqe_prev != \
            &(elm)->field.tqe_next) \
        QUEUEDEBUG_ABORT("TAILQ_* forw %p %s:%d", (elm), \
            __FILE__, __LINE__); \
    if (*(elm)->field.tqe_prev != (elm)) \
        QUEUEDEBUG_ABORT("TAILQ_* back %p %s:%d", (elm), \
            __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_PREREMOVE(head, elm, field) \
    if ((elm)->field.tqe_next == NULL && \
        (head)->tqh_last != &(elm)->field.tqe_next) \
        QUEUEDEBUG_ABORT("TAILQ_PREREMOVE head %p elm %p %s:%d", \
            (head), (elm), __FILE__, \
            __LINE__);
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field) \
    (elm)->field.tqe_next = (void *)1L; \
    (elm)->field.tqe_prev = (void *)1L;
#else
#define QUEUEDEBUG_TAILQ_INSERT_HEAD(head, elm, field)
#define QUEUEDEBUG_TAILQ_INSERT_TAIL(head, elm, field)
#define QUEUEDEBUG_TAILQ_OP(elm, field)
#define QUEUEDEBUG_TAILQ_PREREMOVE(head, elm, field)
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field)
#endif

#define TAILQ_INIT(head) do { \
    (head)->tqh_first = TAILQ_END(head); \
    (head)->tqh_last = &(head)->tqh_first; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_HEAD(head, elm, field) do { \
    QUEUEDEBUG_TAILQ_INSERT_HEAD((head), (elm), field) \
    if (((elm)->field.tqe_next = (head)->tqh_first) != TAILQ_END(head)) \
        (head)->tqh_first->field.tqe_prev = \
            &(elm)->field.tqe_next; \
    else \
        (head)->tqh_last = &(elm)->field.tqe_next; \
    (head)->tqh_first = (elm); \
    (elm)->field.tqe_prev = &(head)->tqh_first; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_TAIL(head,

```

```

elm, field) do { \
QUEUEDEBUG_TAILQ_INSERT_TAIL((head), (elm), field) \
(elm)->field.tqe_next = TAILQ_END(head); \
(elm)->field.tqe_prev = (head)->tqh_last; \
*(head)->tqh_last = (elm); \
(head)->tqh_last = &(elm)->field.tqe_next; \
} while (/*CONSTCOND*/0)

```

```

#define TAILQ_INSERT_AFTER(head, listelm, elm, field) do { \
QUEUEDEBUG_TAILQ_OP((listelm), field) \
if (((elm)->field.tqe_next = (listelm)->field.tqe_next) != \
TAILQ_END(head)) \
(elm)->field.tqe_next->field.tqe_prev = \
&(elm)->field.tqe_next; \
else \
(head)->tqh_last = &(elm)->field.tqe_next; \
(listelm)->field.tqe_next = (elm); \
(elm)->field.tqe_prev = &(listelm)->field.tqe_next; \
} while (/*CONSTCOND*/0)

```

```

#define TAILQ_INSERT_BEFORE(listelm, elm, field) do { \
QUEUEDEBUG_TAILQ_OP((listelm), field) \
(elm)->field.tqe_prev = (listelm)->field.tqe_prev; \
(elm)->field.tqe_next = (listelm); \
*(listelm)->field.tqe_prev = (elm); \
(listelm)->field.tqe_prev \
= &(elm)->field.tqe_next; \
} while (/*CONSTCOND*/0)

```

```

#define TAILQ_REMOVE(head, elm, field) do { \
QUEUEDEBUG_TAILQ_PREREMOVE((head), (elm), field) \
QUEUEDEBUG_TAILQ_OP((elm), field) \
if (((elm)->field.tqe_next) != TAILQ_END(head)) \
(elm)->field.tqe_next->field.tqe_prev = \
(elm)->field.tqe_prev; \
else \
(head)->tqh_last = (elm)->field.tqe_prev; \
*(elm)->field.tqe_prev = (elm)->field.tqe_next; \
QUEUEDEBUG_TAILQ_POSTREMOVE((elm), field); \
} while (/*CONSTCOND*/0)

```

```

#define TAILQ_REPLACE(head, elm, elm2, field) do { \
if (((elm2)->field.tqe_next = (elm)->field.tqe_next) != \
TAILQ_END(head)) \
(elm2)->field.tqe_next->field.tqe_prev = \
&(elm2)->field.tqe_next; \
else \
(head)->tqh_last = &(elm2)->field.tqe_next; \
} while (/*CONSTCOND*/0)

```

```

        (elm2)->field.tqe_prev = (elm)->field.tqe_prev; \
        *(elm2)->field.tqe_prev = (elm2); \
    QUEUEDEBUG_TAILQ_POSTREMOVE((elm),
    field); \
} while (/*CONSTCOND*/0)

#define TAILQ_CONCAT(head1, head2, field) do { \
    if (!TAILQ_EMPTY(head2)) { \
        *(head1)->tqh_last = (head2)->tqh_first; \
        (head2)->tqh_first->field.tqe_prev = (head1)->tqh_last; \
        (head1)->tqh_last = (head2)->tqh_last; \
        TAILQ_INIT((head2)); \
    } \
} while (/*CONSTCOND*/0)

/*
 * Singly-linked Tail queue declarations.
 */
#define STAILQ_HEAD(name, type) \
struct name { \
    struct type *stqh_first; /* first element */ \
    struct type **stqh_last; /* addr of last next element */ \
}

#define STAILQ_HEAD_INITIALIZER(head) \
{ NULL, &(head).stqh_first }

#define STAILQ_ENTRY(type) \
struct { \
    struct type *stqe_next; /* next element */ \
}

/*
 * Singly-linked Tail queue access methods.
 */
#define STAILQ_FIRST(head) ((head)->stqh_first)
#define STAILQ_END(head) NULL
#define STAILQ_NEXT(elm, field) ((elm)->field.stqe_next)
#define STAILQ_EMPTY(head) (STAILQ_FIRST(head) \
    == STAILQ_END(head))

/*
 * Singly-linked Tail queue functions.
 */
#define STAILQ_INIT(head) do { \
    (head)->stqh_first = NULL; \
    (head)->stqh_last = &(head)->stqh_first; \
} while (/*CONSTCOND*/0)

```

```

#define STAILQ_INSERT_HEAD(head, elm, field) do { \
    if (((elm)->field.stqe_next = (head)->stqh_first) == NULL) \
        (head)->stqh_last = &(elm)->field.stqe_next; \
    (head)->stqh_first = (elm); \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_TAIL(head, elm, field) do { \
    (elm)->field.stqe_next = NULL; \
    *(head)->stqh_last = (elm); \
    (head)->stqh_last = &(elm)->field.stqe_next; \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_AFTER(head, listelm, elm, field) do { \
    if (((elm)->field.stqe_next = (listelm)->field.stqe_next) == NULL) \
        (head)->stqh_last = &(elm)->field.stqe_next; \
    (listelm)->field.stqe_next = (elm); \
} while (/*CONSTCOND*/0)

#define STAILQ_REMOVE_HEAD(head, field) do { \
    if (((head)->stqh_first = (head)->stqh_first->field.stqe_next) \
        == NULL) \
        (head)->stqh_last = &(head)->stqh_first; \
} while (/*CONSTCOND*/0)

#define STAILQ_REMOVE(head, elm, type, field) do { \
    if ((head)->stqh_first == (elm)) { \
        STAILQ_REMOVE_HEAD((head), field); \
    } else { \
        struct type *curelm = (head)->stqh_first; \
        while (curelm->field.stqe_next != (elm)) \
            curelm = curelm->field.stqe_next; \
        if ((curelm->field.stqe_next = \
            curelm->field.stqe_next->field.stqe_next) == NULL) \
            (head)->stqh_last = &(curelm)->field.stqe_next; \
    } \
} while (/*CONSTCOND*/0)

#define STAILQ_FOREACH(var, head, field) \
    for ((var) = ((head)->stqh_first); \
        (var); \
        (var) = ((var)->field.stqe_next))

#define STAILQ_FOREACH_SAFE(var, head, field, tvar) \
    for ((var) = STAILQ_FIRST((head)); \
        (var) && ((tvar) = STAILQ_NEXT((var), field), 1); \
        (var) = (tvar))

```

```

#define STAILQ_CONCAT(head1, head2) do { \
    if (!STAILQ_EMPTY((head2))) { \
        *(head1)->stqh_last \
        = (head2)->stqh_first; \
        (head1)->stqh_last = (head2)->stqh_last; \
        STAILQ_INIT((head2)); \
    } \
} while (/*CONSTCOND*/0)

#define STAILQ_LAST(head, type, field) \
(STAILQ_EMPTY((head)) ? \
    NULL : \
    ((struct type *) (void *) \
    ((char *)((head)->stqh_last) - offsetof(struct type, field))))

#ifdef _KERNEL
/*
 * Circular queue definitions. Do not use. We still keep the macros
 * for compatibility but because of pointer aliasing issues their use
 * is discouraged!
 */

/*
 * __launder_type(): We use this ugly hack to work around the the compiler
 * noticing that two types may not alias each other and elide tests in code.
 * We hit this in the CIRCLEQ macros when comparing 'struct name *' and
 * 'struct type *' (see CIRCLEQ_HEAD()). Modern compilers (such as GCC
 * 4.8) declare these comparisons as always false, causing the code to
 * not run as designed.
 *
 * This hack is only to be used for comparisons and thus can be fully const.
 * Do not use for assignment.
 *
 * If we ever choose to change the ABI of the CIRCLEQ macros, we could fix
 * this by changing the head/tail sentinel values, but see the note above
 * this one.
 */
static __inline const void * __launder_type(const void *);
static __inline const void *
__launder_type(const void *__x)
{
    __asm __volatile("" : "+r" (__x));
    return __x;
}

#ifdef QUEUEDEBDEBUG
#define QUEUEDEBDEBUG_CIRCLEQ_HEAD(head, field) \

```



```

if ((head)->cqh_first != CIRCLEQ_ENDC(head) && \
    (head)->cqh_first->field.cqe_prev != CIRCLEQ_ENDC(head)) \
    QUEUEDEBDEBUG_ABORT("CIRCLEQ head forw %p %s:%d", (head), \
        __FILE__, __LINE__); \
if ((head)->cqh_last != CIRCLEQ_ENDC(head) && \
    (head)->cqh_last->field.cqe_next != CIRCLEQ_ENDC(head)) \
    QUEUEDEBDEBUG_ABORT("CIRCLEQ head back %p %s:%d", (head), \
        __FILE__, __LINE__);
#define QUEUEDEBDEBUG_CIRCLEQ_ELM(head, elm, field) \
if ((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) { \
    if ((head)->cqh_last != (elm)) \
        QUEUEDEBDEBUG_ABORT("CIRCLEQ \
elm last %p %s:%d", \
            (elm), __FILE__, __LINE__); \
} else { \
    if ((elm)->field.cqe_next->field.cqe_prev != (elm)) \
        QUEUEDEBDEBUG_ABORT("CIRCLEQ elm forw %p %s:%d", \
            (elm), __FILE__, __LINE__); \
} \
if ((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) { \
    if ((head)->cqh_first != (elm)) \
        QUEUEDEBDEBUG_ABORT("CIRCLEQ elm first %p %s:%d", \
            (elm), __FILE__, __LINE__); \
} else { \
    if ((elm)->field.cqe_prev->field.cqe_next != (elm)) \
        QUEUEDEBDEBUG_ABORT("CIRCLEQ elm prev %p %s:%d", \
            (elm), __FILE__, __LINE__); \
}
#define QUEUEDEBDEBUG_CIRCLEQ_POSTREMOVE(elm, field) \
(elm)->field.cqe_next = (void *)1L; \
(elm)->field.cqe_prev = (void *)1L;
#else
#define QUEUEDEBDEBUG_CIRCLEQ_HEAD(head, field)
#define QUEUEDEBDEBUG_CIRCLEQ_ELM(head, elm, field)
#define QUEUEDEBDEBUG_CIRCLEQ_POSTREMOVE(elm, field)
#endif

#define CIRCLEQ_HEAD(name, type) \
struct name { \
    struct type *cqh_first; /* \
first element */ \
    struct type *cqh_last; /* last element */ \
}

#define CIRCLEQ_HEAD_INITIALIZER(head) \
{ CIRCLEQ_END(&head), CIRCLEQ_END(&head) }

#define CIRCLEQ_ENTRY(type) \

```

```

struct {
    \
    struct type *cqe_next; /* next element */ \
    struct type *cqe_prev; /* previous element */ \
}

/*
 * Circular queue functions.
 */

#define CIRCLEQ_INIT(head) do { \
    (head)->cqh_first = CIRCLEQ_END(head); \
    (head)->cqh_last = CIRCLEQ_END(head); \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_AFTER(head, listelm, elm, field) do { \
    QUEUEDEBBUG_CIRCLEQ_HEAD((head), field) \
    QUEUEDEBBUG_CIRCLEQ_ELM((head), (listelm), field) \
    (elm)->field.cqe_next = (listelm)->field.cqe_next; \
    (elm)->field.cqe_prev = (listelm); \
    if ((listelm)->field.cqe_next == CIRCLEQ_ENDC(head)) \
        (head)->cqh_last = (elm); \
    else \
        (listelm)->field.cqe_next->field.cqe_prev = (elm); \
    (listelm)->field.cqe_next = (elm); \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_BEFORE(head, \
listelm, elm, field) do { \
    QUEUEDEBBUG_CIRCLEQ_HEAD((head), field) \
    QUEUEDEBBUG_CIRCLEQ_ELM((head), (listelm), field) \
    (elm)->field.cqe_next = (listelm); \
    (elm)->field.cqe_prev = (listelm)->field.cqe_prev; \
    if ((listelm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \
        (head)->cqh_first = (elm); \
    else \
        (listelm)->field.cqe_prev->field.cqe_next = (elm); \
    (listelm)->field.cqe_prev = (elm); \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_HEAD(head, elm, field) do { \
    QUEUEDEBBUG_CIRCLEQ_HEAD((head), field) \
    (elm)->field.cqe_next = (head)->cqh_first; \
    (elm)->field.cqe_prev = CIRCLEQ_END(head); \
    if ((head)->cqh_last == CIRCLEQ_ENDC(head)) \
        (head)->cqh_last = (elm); \
    else \
        (head)->cqh_first->field.cqe_prev = (elm); \
    (head)->cqh_first = (elm); \
} while (/*CONSTCOND*/0)

```

```

#define CIRCLEQ_INSERT_TAIL(head, elm, field) do { \
    QUEUEDEBUG_CIRCLEQ_HEAD((head), field) \
    (elm)->field.cqe_next \
    = CIRCLEQ_END(head); \
    (elm)->field.cqe_prev = (head)->cqh_last; \
    if ((head)->cqh_first == CIRCLEQ_ENDC(head)) \
        (head)->cqh_first = (elm); \
    else \
        (head)->cqh_last->field.cqe_next = (elm); \
    (head)->cqh_last = (elm); \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_REMOVE(head, elm, field) do { \
    QUEUEDEBUG_CIRCLEQ_HEAD((head), field) \
    QUEUEDEBUG_CIRCLEQ_ELM((head), (elm), field) \
    if ((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) \
        (head)->cqh_last = (elm)->field.cqe_prev; \
    else \
        (elm)->field.cqe_next->field.cqe_prev = \
            (elm)->field.cqe_prev; \
    if ((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \
        (head)->cqh_first = (elm)->field.cqe_next; \
    else \
        (elm)->field.cqe_prev->field.cqe_next = \
            (elm)->field.cqe_next; \
    QUEUEDEBUG_CIRCLEQ_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_FOREACH(var, head, field) \
    for ((var) = ((head)->cqh_first); \
        (var) != CIRCLEQ_ENDC(head); \
        (var) \
        = ((var)->field.cqe_next))

#define CIRCLEQ_FOREACH_REVERSE(var, head, field) \
    for ((var) = ((head)->cqh_last); \
        (var) != CIRCLEQ_ENDC(head); \
        (var) = ((var)->field.cqe_prev))

/*
 * Circular queue access methods.
 */

#define CIRCLEQ_FIRST(head) ((head)->cqh_first)
#define CIRCLEQ_LAST(head) ((head)->cqh_last)
/* For comparisons */
#define CIRCLEQ_ENDC(head) (__launter_type(head))
/* For assignments */

```

```

#define CIRCLEQ_END(head) ((void *)(head))
#define CIRCLEQ_NEXT(elm, field) ((elm)->field.cqe_next)
#define CIRCLEQ_PREV(elm, field) ((elm)->field.cqe_prev)
#define CIRCLEQ_EMPTY(head) \
    (CIRCLEQ_FIRST(head) == CIRCLEQ_ENDC(head))

#define CIRCLEQ_LOOP_NEXT(head, elm, field) \
    (((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) \
     ? ((head)->cqh_first) \
     : (elm->field.cqe_next))
#define CIRCLEQ_LOOP_PREV(head, elm, field) \
    (((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \
     ? ((head)->cqh_last) \
     : \
     (elm->field.cqe_prev))
#endif /* !_KERNEL */

#endif /* !_SYS_QUEUE_H_ */
#ifndef _SYS_CDEFS_H_
#define _SYS_CDEFS_H_

#warning usage of non-standard #include <sys/cdefs.h> is deprecated

#undef __P
#undef __PMT

#define __P(args) args
#define __PMT(args) args

#define __CONCAT(x,y) x ## y
#define __STRING(x) #x

#ifdef __cplusplus
# define __BEGIN_DECLS extern "C" {
# define __END_DECLS }
#else
# define __BEGIN_DECLS
# define __END_DECLS
#endif

#if defined(__GNUC__) && !defined(__cplusplus)
# define __THROW __attribute__((__nothrow__))
# define __NTH(fct) __attribute__((__nothrow__)) fct
#else
# define __THROW
# define __NTH(fct) fct
#endif

```

```

#endif /* _SYS_CDEFS_H_ */
# Contributor: Fabian Affolter <fabian@affolter-engineering.ch>
# Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>
pkgname=py3-flake8-copyright
_pkgname=flake8-copyright
pkgver=0.2.4
pkgrel=3
pkgdesc="Extension for flake8 which checks for copyrights"
options="!check" # No testsuite
url="https://github.com/savoirfairelinux/flake8-copyright"
arch="noarch"
license="MIT"
depends="py3-flake8 py3-setuptools"
makedepends="py3-gpep517 py3-wheel"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/${_pkgname:0:1}/${_pkgname}/${_pkgname}-${pkgver}.tar.gz"
buildid="srcdir"/$_pkgname-$pkgver

replaces="py-flake8-copyright" # Backwards compatibility
provides="py-flake8-copyright=${pkgver}-r${pkgrel}" # Backwards compatibility

build() {
    gpep517 build-wheel \
        --wheel-dir .dist \
        --output-fd 3 3>&1 >&2
}

package() {
    gpep517 install-wheel --destdir "$pkgdir" \
        .dist/*.whl
}

sha512sums="
a6ab47e1bb715618f075c51f398e18180404871b3b0faf1c2d30701d5203db6bc23771bf22ffb5bafcd8ee856b9b64237
b316bd503ae7dceed6ca284ccd5a74d
    flake8-copyright-0.2.4.tar.gz
"

# Maintainer: fossdd <fossdd@pwned.life>
pkgname=cargo-license
pkgver=0.6.1
pkgrel=1
pkgdesc="Cargo subcommand to see license of dependencies"
url="https://github.com/onur/cargo-license"
arch="all"
license="MIT"
makedepends="cargo cargo-auditable"
source="$pkgname-$pkgver.tar.gz::https://crates.io/api/v1/crates/cargo-license/$pkgver/download"

```

```

prepare() {
    default_prepare

    cargo fetch --target="$CTARGET" --locked
}

build() {
    cargo auditable build --release --frozen
}

check() {
    cargo test --frozen
}

package() {
    install -Dm755 target/release/cargo-license -t "$pkgdir"/usr/bin
}

sha512sums="
36215145e79965be090ae2f5ce6f5eb208465eb67a94c26bed1fe70a131c46c6dc35695e30852759b7366883fbc3caf00
272d796ac634f0a5e1d49232b6b29d5  cargo-license-0.6.1.tar.gz
"

/* $NetBSD: tree.h,v 1.20 2013/09/14 13:20:45 joerg Exp $ */
/* $OpenBSD: tree.h,v 1.13 2011/07/09 00:19:45 pirofti Exp $ */
/* Modified by Void Linux. */
/*
* Copyright 2002 Niels Provos <provos@citi.umich.edu>
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE
* AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF

```

\* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

\*/

#ifndef \_SYS\_TREE\_H\_

#define \_SYS\_TREE\_H\_

#ifdef \_\_GNUC\_\_

#define \_\_GNUC\_PREREQ\_\_(x, y) \\\n ((\_\_GNUC\_\_ == (x) && \_\_GNUC\_MINOR\_\_ >= (y)) || \\\n (\_\_GNUC\_\_ > (x)))

#else

#define \_\_GNUC\_PREREQ\_\_(x, y) 0

#endif

#if \_\_GNUC\_PREREQ\_\_(2, 7) || defined(\_\_lint\_\_)

#define \_sys\_tree\_h\_unused \_\_attribute\_\_((\_\_unused\_\_))

#else

#define \_sys\_tree\_h\_unused /\* delete \*/

#endif

/\*

\* This file defines data structures for different types of trees:

\* splay trees and red-black trees.

\*

\* A splay tree is a self-organizing

data structure. Every operation

\* on the tree causes a splay to happen. The splay moves the requested

\* node to the root of the tree and partly rebalances it.

\*

\* This has the benefit that request locality causes faster lookups as

\* the requested nodes move to the top of the tree. On the other hand,

\* every lookup causes memory writes.

\*

\* The Balance Theorem bounds the total access time for m operations

\* and n inserts on an initially empty tree as  $O((m + n) \lg n)$ . The

\* amortized cost for a sequence of m accesses to a splay tree is  $O(\lg n)$ ;

\*

\* A red-black tree is a binary search tree with the node color as an

\* extra attribute. It fulfills a set of conditions:

\* - every search path from the root to a leaf consists of the

\* same number of black nodes,

\* - each red node (except for the root) has a black parent,

\* - each leaf node is black.

\*

\* Every operation on a red-black tree is bounded as  $O(\lg n)$ .

\* The maximum height of a red-black tree is  $2 \lg (n+1)$ .

\*/

```

#define SPLAY_HEAD(name, type)  \
struct name {  \
    struct type *sph_root; /* root of the tree */  \
}

#define SPLAY_INITIALIZER(root)  \
{ NULL }

#define SPLAY_INIT(root) do {  \
    (root)->sph_root = NULL;  \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ENTRY(type)  \
struct {  \
    struct type *spe_left; /* left element */  \
    struct type *spe_right; /* right element */  \
}

#define SPLAY_LEFT(elm, field) (elm)->field.spe_left
#define SPLAY_RIGHT(elm, field) (elm)->field.spe_right
#define SPLAY_ROOT(head) (head)->sph_root
#define SPLAY_EMPTY(head) (SPLAY_ROOT(head) == NULL)

/* SPLAY_ROTATE_{LEFT,RIGHT} expect that tmp hold SPLAY_{RIGHT,LEFT} */
#define SPLAY_ROTATE_RIGHT(head, tmp, field) do {  \
    SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(tmp, field);  \
    SPLAY_RIGHT(tmp, field) = (head)->sph_root;  \
    (head)->sph_root = tmp;  \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ROTATE_LEFT(head, tmp, field) do {  \
    SPLAY_RIGHT((head)->sph_root,  \
    field) = SPLAY_LEFT(tmp, field);  \
    SPLAY_LEFT(tmp, field) = (head)->sph_root;  \
    (head)->sph_root = tmp;  \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKLEFT(head, tmp, field) do {  \
    SPLAY_LEFT(tmp, field) = (head)->sph_root;  \
    tmp = (head)->sph_root;  \
    (head)->sph_root = SPLAY_LEFT((head)->sph_root, field);  \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKRIGHT(head, tmp, field) do {  \
    SPLAY_RIGHT(tmp, field) = (head)->sph_root;  \
    tmp = (head)->sph_root;  \
    (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field);  \
} while (/*CONSTCOND*/ 0)

```



```

#define SPLAY_ASSEMBLE(head, node, left, right, field) do { \
    SPLAY_RIGHT(left, field) = SPLAY_LEFT((head)->sph_root, field); \
    SPLAY_LEFT(right, field) = SPLAY_RIGHT((head)->sph_root, field); \
    SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(node, field); \
    SPLAY_RIGHT((head)->sph_root, field) = SPLAY_LEFT(node, field); \
} while (/*CONSTCOND*/ 0)

```

/\* Generates prototypes and inline functions \*/

```

#define
    SPLAY_PROTOTYPE(name, type, field, cmp) \
void name##_SPLAY(struct name *, struct type *); \
void name##_SPLAY_MINMAX(struct name *, int); \
struct type *name##_SPLAY_INSERT(struct name *, struct type *); \
struct type *name##_SPLAY_REMOVE(struct name *, struct type *); \
    \
/* Finds the node with the same key as elm */ \
static __inline struct type * \
name##_SPLAY_FIND(struct name *head, struct type *elm) \
{ \
    if (SPLAY_EMPTY(head)) \
        return(NULL); \
    name##_SPLAY(head, elm); \
    if ((cmp)(elm, (head)->sph_root) == 0) \
        return (head->sph_root); \
    return (NULL); \
} \
    \
static __inline _sys_tree_h_unused struct type * \
name##_SPLAY_NEXT(struct name *head, struct type *elm) \
{ \
    name##_SPLAY(head, elm); \
    if (SPLAY_RIGHT(elm, field) != NULL) { \
        elm = SPLAY_RIGHT(elm, field); \
        while (SPLAY_LEFT(elm, field) != NULL) { \
            elm = SPLAY_LEFT(elm, field); \
        } \
    } \
    else \
        elm = NULL; \
    return (elm); \
} \
    \
static _sys_tree_h_unused __inline struct type * \
name##_SPLAY_MIN_MAX(struct name *head, int val) \
{ \
    name##_SPLAY_MINMAX(head, val); \
}

```

```

    return (SPLAY_ROOT(head)); \
}

/* Main splay operation.
 * Moves node close to the key of elm to top
 */
#define SPLAY_GENERATE(name, type, field, cmp) \
struct type * \
name##_SPLAY_INSERT(struct name *head, struct type *elm) \
{ \
    if (SPLAY_EMPTY(head)) { \
        SPLAY_LEFT(elm, field) = SPLAY_RIGHT(elm, field) = NULL; \
    } else { \
        int __comp; \
        name##_SPLAY(head, elm); \
        __comp = (cmp)(elm, (head)->sph_root); \
        if (__comp < 0) { \
            SPLAY_LEFT(elm, field) = SPLAY_LEFT((head)->sph_root, field); \
            SPLAY_RIGHT(elm, field) = (head)->sph_root; \
            SPLAY_LEFT((head)->sph_root, field) = NULL; \
        } else if (__comp > 0) { \
            SPLAY_RIGHT(elm, \
field) = SPLAY_RIGHT((head)->sph_root, field); \
            SPLAY_LEFT(elm, field) = (head)->sph_root; \
            SPLAY_RIGHT((head)->sph_root, field) = NULL; \
        } else \
            return ((head)->sph_root); \
    } \
    (head)->sph_root = (elm); \
    return (NULL); \
} \
\
struct type * \
name##_SPLAY_REMOVE(struct name *head, struct type *elm) \
{ \
    struct type *__tmp; \
    if (SPLAY_EMPTY(head)) \
        return (NULL); \
    name##_SPLAY(head, elm); \
    if ((cmp)(elm, (head)->sph_root) == 0) { \
        if (SPLAY_LEFT((head)->sph_root, field) == NULL) { \
            (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field); \
        } else { \
            __tmp = SPLAY_RIGHT((head)->sph_root, field); \
            (head)->sph_root = SPLAY_LEFT((head)->sph_root, field); \
            name##_SPLAY(head, elm); \
            SPLAY_RIGHT((head)->sph_root, field) = __tmp; \
        } \
    } \
} \

```

```

    return (elm);    \
}    \
return (NULL);    \
}    \
    \
void    \
name##_SPLAY(struct
name *head, struct type *elm) \
{    \
    struct type __node, *__left, *__right, *__tmp; \
    int __comp;    \
    \
    SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
    __left = __right = &__node;    \
    \
    while ((__comp = (cmp)(elm, (head)->sph_root)) != 0) { \
        if (__comp < 0) { \
            __tmp = SPLAY_LEFT((head)->sph_root, field); \
            if (__tmp == NULL) \
                break; \
            if ((cmp)(elm, __tmp) < 0){ \
                SPLAY_ROTATE_RIGHT(head, __tmp, field); \
                if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
                    break; \
            } \
            SPLAY_LINKLEFT(head, __right, field); \
        } else if (__comp > 0) { \
            __tmp = SPLAY_RIGHT((head)->sph_root, field); \
            if (__tmp == NULL) \
                break; \
            if ((cmp)(elm, __tmp) > 0){ \
                SPLAY_ROTATE_LEFT(head, __tmp, field); \
                if (SPLAY_RIGHT((head)->sph_root, field) == NULL)\
                    break; \
            } \
            SPLAY_LINKRIGHT(head, __left, field); \
        } \
    } \
    SPLAY_ASSEMBLE(head,
&__node, __left, __right, field); \
}    \
    \
/* Splay with either the minimum or the maximum element \
* Used to find minimum or maximum element in tree. \
*/    \
void name##_SPLAY_MINMAX(struct name *head, int __comp) \
{    \
    struct type __node, *__left, *__right, *__tmp; \

```

```

\
SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
__left = __right = &__node; \
\
while (1) { \
    if (__comp < 0) { \
        __tmp = SPLAY_LEFT((head)->sph_root, field);\
        if (__tmp == NULL) \
            break; \
        if (__comp < 0){ \
            SPLAY_ROTATE_RIGHT(head, __tmp, field);\
            if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
                break; \
        } \
        SPLAY_LINKLEFT(head, __right, field); \
    } else if (__comp > 0) { \
        __tmp = SPLAY_RIGHT((head)->sph_root, field);\
        if (__tmp == NULL) \
            break; \
        if (__comp > 0) { \
            SPLAY_ROTATE_LEFT(head, __tmp, field);\
            if (SPLAY_RIGHT((head)->sph_root,
field) == NULL)\
                break; \
        } \
        SPLAY_LINKRIGHT(head, __left, field); \
    } \
} \
SPLAY_ASSEMBLE(head, &__node, __left, __right, field); \
}

#define SPLAY_NEGINF -1
#define SPLAY_INF 1

#define SPLAY_INSERT(name, x, y) name##_SPLAY_INSERT(x, y)
#define SPLAY_REMOVE(name, x, y) name##_SPLAY_REMOVE(x, y)
#define SPLAY_FIND(name, x, y) name##_SPLAY_FIND(x, y)
#define SPLAY_NEXT(name, x, y) name##_SPLAY_NEXT(x, y)
#define SPLAY_MIN(name, x) (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_NEGINF))
#define SPLAY_MAX(name, x) (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_INF))

#define SPLAY_FOREACH(x, name, head) \
    for ((x) = SPLAY_MIN(name, head); \
        (x) != NULL; \
        (x) = SPLAY_NEXT(name, head, x))

```

```

/* Macros that define a red-black tree */
#define RB_HEAD(name, type) \
struct name { \
    struct type *rbh_root; /* root of the tree */ \
}

#define RB_INITIALIZER(root) \
{ \
    NULL \
}

#define RB_INIT(root) do { \
    (root)->rbh_root = NULL; \
} while (/*CONSTCOND*/ 0)

#define RB_BLACK 0
#define RB_RED 1
#define RB_ENTRY(type) \
struct { \
    struct type *rbe_left; /* left element */ \
    struct type *rbe_right; /* right element */ \
    struct type *rbe_parent; /* parent element */ \
    int rbe_color; /* node color */ \
}

#define RB_LEFT(elm, field) (elm)->field.rbe_left
#define RB_RIGHT(elm, field) (elm)->field.rbe_right
#define RB_PARENT(elm, field) (elm)->field.rbe_parent
#define RB_COLOR(elm, field) (elm)->field.rbe_color
#define RB_ROOT(head) (head)->rbh_root
#define RB_EMPTY(head) (RB_ROOT(head) == NULL)

#define RB_SET(elm, parent, field) do { \
    RB_PARENT(elm, field) = parent; \
    RB_LEFT(elm, field) = RB_RIGHT(elm, field) = NULL; \
    RB_COLOR(elm, field) = RB_RED; \
} while (/*CONSTCOND*/ 0)

#define RB_SET_BLACKRED(black, red, field) do { \
    RB_COLOR(black, field) = RB_BLACK; \
    RB_COLOR(red, field) \
    = RB_RED; \
} while (/*CONSTCOND*/ 0)

#ifndef RB_AUGMENT
#define RB_AUGMENT(x) do {} while (/*CONSTCOND*/ 0)
#endif

#define RB_ROTATE_LEFT(head, elm, tmp, field) do { \

```

```

(tmp) = RB_RIGHT(elm, field); \
if ((RB_RIGHT(elm, field) = RB_LEFT(tmp, field)) != NULL) { \
    RB_PARENT(RB_LEFT(tmp, field), field) = (elm); \
} \
RB_AUGMENT(elm); \
if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
    if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
        RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
    else \
        RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
} else \
    (head)->rbh_root = (tmp); \
RB_LEFT(tmp, field) = (elm); \
RB_PARENT(elm, field) = (tmp); \
RB_AUGMENT(tmp); \
if ((RB_PARENT(tmp, field))) \
    RB_AUGMENT(RB_PARENT(tmp, field)); \
} while (/*CONSTCOND*/ 0)

#define RB_ROTATE_RIGHT(head, elm, tmp, field) do { \
    (tmp) = RB_LEFT(elm, field); \
    if ((RB_LEFT(elm, field) = RB_RIGHT(tmp, field)) \
        != NULL) { \
        RB_PARENT(RB_RIGHT(tmp, field), field) = (elm); \
    } \
    RB_AUGMENT(elm); \
    if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
        if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
            RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
        else \
            RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
    } else \
        (head)->rbh_root = (tmp); \
    RB_RIGHT(tmp, field) = (elm); \
    RB_PARENT(elm, field) = (tmp); \
    RB_AUGMENT(tmp); \
    if ((RB_PARENT(tmp, field))) \
        RB_AUGMENT(RB_PARENT(tmp, field)); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */
#define RB_PROTOTYPE(name, type, field, cmp) \
    RB_PROTOTYPE_INTERNAL(name, type, field, cmp,)
#define RB_PROTOTYPE_STATIC(name, type, field, cmp) \
    RB_PROTOTYPE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_PROTOTYPE_INTERNAL(name, type, field, cmp, attr) \
    attr void name##_RB_INSERT_COLOR(struct name *, struct

```

```

type *); \
attr void name##_RB_REMOVE_COLOR(struct name *, struct type *, struct type *);\
attr struct type *name##_RB_REMOVE(struct name *, struct type *); \
attr struct type *name##_RB_INSERT(struct name *, struct type *); \
attr struct type *name##_RB_FIND(struct name *, struct type *); \
attr struct type *name##_RB_NFIND(struct name *, struct type *); \
attr struct type *name##_RB_NEXT(struct type *); \
attr struct type *name##_RB_PREV(struct type *); \
attr struct type *name##_RB_MINMAX(struct name *, int); \
\

/* Main rb operation.
* Moves node close to the key of elm to top
*/
#define RB_GENERATE(name, type, field, cmp) \
    RB_GENERATE_INTERNAL(name, type, field, cmp,)
#define RB_GENERATE_STATIC(name, type, field, cmp) \
    RB_GENERATE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_GENERATE_INTERNAL(name, type, field, cmp, attr) \
attr void \
name##_RB_INSERT_COLOR(struct name *head, struct type *elm) \
{ \
    struct type \
    *parent, *gparent, *tmp; \
    while ((parent = RB_PARENT(elm, field)) != NULL && \
        RB_COLOR(parent, field) == RB_RED) { \
        gparent = RB_PARENT(parent, field); \
        if (parent == RB_LEFT(gparent, field)) { \
            tmp = RB_RIGHT(gparent, field); \
            if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
                RB_COLOR(tmp, field) = RB_BLACK; \
                RB_SET_BLACKRED(parent, gparent, field);\
                elm = gparent; \
                continue; \
            } \
            if (RB_RIGHT(parent, field) == elm) { \
                RB_ROTATE_LEFT(head, parent, tmp, field);\
                tmp = parent; \
                parent = elm; \
                elm = tmp; \
            } \
            RB_SET_BLACKRED(parent, gparent, field); \
            RB_ROTATE_RIGHT(head, gparent, tmp, field); \
        } else { \
            tmp = RB_LEFT(gparent, field); \
            if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
                RB_COLOR(tmp, field) = RB_BLACK; \
                RB_SET_BLACKRED(parent, gparent, field);\

```

```

    elm = gparent; \
    continue; \
} \
if (RB_LEFT(parent, field) == elm)
{ \
    RB_ROTATE_RIGHT(head, parent, tmp, field);\
    tmp = parent; \
    parent = elm; \
    elm = tmp; \
} \
RB_SET_BLACKRED(parent, gparent, field); \
RB_ROTATE_LEFT(head, gparent, tmp, field); \
} \
} \
RB_COLOR(head->rbh_root, field) = RB_BLACK; \
} \
\
attr void \
name##_RB_REMOVE_COLOR(struct name *head, struct type *parent, struct type *elm) \
{ \
    struct type *tmp; \
    while ((elm == NULL || RB_COLOR(elm, field) == RB_BLACK) && \
        elm != RB_ROOT(head)) { \
        if (RB_LEFT(parent, field) == elm) { \
            tmp = RB_RIGHT(parent, field); \
            if (RB_COLOR(tmp, field) == RB_RED) { \
                RB_SET_BLACKRED(tmp, parent, field); \
                RB_ROTATE_LEFT(head, parent, tmp, field);\
                tmp = RB_RIGHT(parent, field); \
            } \
            if ((RB_LEFT(tmp, field) == NULL || \
                RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) && \
                (RB_RIGHT(tmp, field) == NULL || \
                RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK)) {\
                RB_COLOR(tmp, field) = RB_RED; \
                elm = parent; \
                parent = RB_PARENT(elm, field); \
            } else { \
                if (RB_RIGHT(tmp, field) == NULL || \
                    RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK) {\
                    struct type *oleft; \
                    if ((oleft = RB_LEFT(tmp, field)) \
                        != NULL) \
                        RB_COLOR(oleft, field) = RB_BLACK;\
                    RB_COLOR(tmp, field) = RB_RED; \
                    RB_ROTATE_RIGHT(head, tmp, oleft, field);\
                    tmp = RB_RIGHT(parent, field); \

```



```

    } \
    RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
    RB_COLOR(parent, field) = RB_BLACK;\
    if (RB_RIGHT(tmp, field)) \
        RB_COLOR(RB_RIGHT(tmp, field), field) = RB_BLACK;\
    RB_ROTATE_LEFT(head, parent, tmp, field);\
    elm = RB_ROOT(head); \
    break; \
} \
} else { \
    tmp = RB_LEFT(parent, field); \
    if (RB_COLOR(tmp, field) == RB_RED) { \
        RB_SET_BLACKRED(tmp, parent, field);\
        RB_ROTATE_RIGHT(head, parent, tmp, field);\
        tmp
= RB_LEFT(parent, field); \
    } \
    if ((RB_LEFT(tmp, field) == NULL || \
        RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) &&\
        (RB_RIGHT(tmp, field) == NULL || \
        RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK)) {\
        RB_COLOR(tmp, field) = RB_RED; \
        elm = parent; \
        parent = RB_PARENT(elm, field); \
    } else { \
        if (RB_LEFT(tmp, field) == NULL || \
            RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) {\
            struct type *oright; \
            if ((oright = RB_RIGHT(tmp, field)) \
                != NULL) \
                RB_COLOR(oright, field) = RB_BLACK;\
            RB_COLOR(tmp, field) = RB_RED; \
            RB_ROTATE_LEFT(head, tmp, oright, field);\
            tmp = RB_LEFT(parent, field); \
        } \
        RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
        RB_COLOR(parent, field) = RB_BLACK;\
        if (RB_LEFT(tmp, field)) \
            RB_COLOR(RB_LEFT(tmp, field), field) = RB_BLACK;\
        RB_ROTATE_RIGHT(head, parent, tmp, field);\
        elm = RB_ROOT(head); \
        break; \
    } \
} \
} \
if
(elm) \
    RB_COLOR(elm, field) = RB_BLACK; \

```

```

}    \
    \
attr struct type *    \
name##_RB_REMOVE(struct name *head, struct type *elm) \
{    \
    struct type *child, *parent, *old = elm; \
    int color;    \
    if (RB_LEFT(elm, field) == NULL) \
        child = RB_RIGHT(elm, field); \
    else if (RB_RIGHT(elm, field) == NULL) \
        child = RB_LEFT(elm, field); \
    else {    \
        struct type *left; \
        elm = RB_RIGHT(elm, field); \
        while ((left = RB_LEFT(elm, field)) != NULL) \
            elm = left; \
        child = RB_RIGHT(elm, field); \
        parent = RB_PARENT(elm, field); \
        color = RB_COLOR(elm, field); \
        if (child) \
            RB_PARENT(child, field) = parent; \
        if (parent) {    \
            if (RB_LEFT(parent, field) == elm) \
                RB_LEFT(parent, field) = child; \
            else \
                RB_RIGHT(parent, field) = child; \
            RB_AUGMENT(parent); \
        } else \
            RB_ROOT(head)
    = child; \
    if (RB_PARENT(elm, field) == old) \
        parent = elm; \
    (elm)->field = (old)->field; \
    if (RB_PARENT(old, field)) { \
        if (RB_LEFT(RB_PARENT(old, field), field) == old)\
            RB_LEFT(RB_PARENT(old, field), field) = elm;\
        else \
            RB_RIGHT(RB_PARENT(old, field), field) = elm;\
        RB_AUGMENT(RB_PARENT(old, field)); \
    } else \
        RB_ROOT(head) = elm; \
    RB_PARENT(RB_LEFT(old, field), field) = elm; \
    if (RB_RIGHT(old, field)) \
        RB_PARENT(RB_RIGHT(old, field), field) = elm; \
    if (parent) {    \
        left = parent; \
        do {    \
            RB_AUGMENT(left); \

```

```

    } while ((left = RB_PARENT(left, field)) != NULL); \
} \
goto color; \
} \
parent = RB_PARENT(elm, field); \
color = RB_COLOR(elm, field); \
if (child) \
    RB_PARENT(child, field) = parent; \
if (parent) { \
    if (RB_LEFT(parent, field) == elm) \
        RB_LEFT(parent, field) = child; \
    else \
        RB_RIGHT(parent, \
field) = child; \
    RB_AUGMENT(parent); \
} else \
    RB_ROOT(head) = child; \
color: \
if (color == RB_BLACK) \
    name##_RB_REMOVE_COLOR(head, parent, child); \
return (old); \
} \
\
/* Inserts a node into the RB tree */ \
attr struct type * \
name##_RB_INSERT(struct name *head, struct type *elm) \
{ \
    struct type *tmp; \
    struct type *parent = NULL; \
    int comp = 0; \
    tmp = RB_ROOT(head); \
    while (tmp) { \
        parent = tmp; \
        comp = (cmp)(elm, parent); \
        if (comp < 0) \
            tmp = RB_LEFT(tmp, field); \
        else if (comp > 0) \
            tmp = RB_RIGHT(tmp, field); \
        else \
            return (tmp); \
    } \
    RB_SET(elm, parent, field); \
    if (parent != NULL) { \
        if (comp < 0) \
            RB_LEFT(parent, field) = elm; \
        else \
            RB_RIGHT(parent, field) = elm; \
        RB_AUGMENT(parent); \
    }

```

```

} else \
    RB_ROOT(head) = elm; \
name##_RB_INSERT_COLOR(head,
elm); \
return (NULL); \
} \
\
/* Finds the node with the same key as elm */ \
attr struct type * \
name##_RB_FIND(struct name *head, struct type *elm) \
{ \
    struct type *tmp = RB_ROOT(head); \
    int comp; \
    while (tmp) { \
        comp = cmp(elm, tmp); \
        if (comp < 0) \
            tmp = RB_LEFT(tmp, field); \
        else if (comp > 0) \
            tmp = RB_RIGHT(tmp, field); \
        else \
            return (tmp); \
    } \
    return (NULL); \
} \
\
/* Finds the first node greater than or equal to the search key */ \
attr struct type * \
name##_RB_NFIND(struct name *head, struct type *elm) \
{ \
    struct type *tmp = RB_ROOT(head); \
    struct type *res = NULL; \
    int comp; \
    while (tmp) { \
        comp = cmp(elm, tmp); \
        if (comp < 0) { \
            res = tmp; \
            tmp = RB_LEFT(tmp, field); \
        } \
        else if (comp > 0) \
            tmp = \
RB_RIGHT(tmp, field); \
        else \
            return (tmp); \
    } \
    return (res); \
} \
\
/* ARGSUSED */ \

```

```

attr struct type *    \
name##_RB_NEXT(struct type *elm)    \
{    \
if (RB_RIGHT(elm, field)) {    \
elm = RB_RIGHT(elm, field);    \
while (RB_LEFT(elm, field))    \
elm = RB_LEFT(elm, field);    \
} else {    \
if (RB_PARENT(elm, field) &&    \
(elm == RB_LEFT(RB_PARENT(elm, field), field))) \
elm = RB_PARENT(elm, field);    \
else {    \
while (RB_PARENT(elm, field) &&    \
(elm == RB_RIGHT(RB_PARENT(elm, field), field)))\
elm = RB_PARENT(elm, field);    \
elm = RB_PARENT(elm, field);    \
}    \
}    \
return (elm);    \
}    \
\
/* ARGSUSED */    \
attr struct type *    \
name##_RB_PREV(struct type *elm)    \
{    \
if (RB_LEFT(elm, field)) {    \
elm = RB_LEFT(elm, field);    \
while (RB_RIGHT(elm, field))    \
elm = RB_RIGHT(elm, field);    \
}    \
else {    \
if (RB_PARENT(elm, field) &&    \
(elm == RB_RIGHT(RB_PARENT(elm, field), field))) \
elm = RB_PARENT(elm, field);    \
else {    \
while (RB_PARENT(elm, field) &&    \
(elm == RB_LEFT(RB_PARENT(elm, field), field)))\
elm = RB_PARENT(elm, field);    \
elm = RB_PARENT(elm, field);    \
}    \
}    \
return (elm);    \
}    \
\
attr struct type *    \
name##_RB_MINMAX(struct name *head, int val)    \
{    \
struct type *tmp = RB_ROOT(head);    \

```

```

struct type *parent = NULL;  \
while (tmp) {  \
    parent = tmp;  \
    if (val < 0)  \
        tmp = RB_LEFT(tmp, field);  \
    else  \
        tmp = RB_RIGHT(tmp, field);  \
}  \
return (parent);  \
}

#define RB_NEGINF -1
#define RB_INF 1

#define RB_INSERT(name, x, y) name##_RB_INSERT(x, y)
#define RB_REMOVE(name, x, y) name##_RB_REMOVE(x, y)
#define RB_FIND(name, x, y) name##_RB_FIND(x, y)
#define RB_NFIND(name, x, y) name##_RB_NFIND(x,
y)
#define RB_NEXT(name, x, y) name##_RB_NEXT(y)
#define RB_PREV(name, x, y) name##_RB_PREV(y)
#define RB_MIN(name, x) name##_RB_MINMAX(x, RB_NEGINF)
#define RB_MAX(name, x) name##_RB_MINMAX(x, RB_INF)

#define RB_FOREACH(x, name, head)  \
for ((x) = RB_MIN(name, head);  \
    (x) != NULL;  \
    (x) = name##_RB_NEXT(x))

#define RB_FOREACH_FROM(x, name, y)  \
for ((x) = (y);  \
    ((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
    (x) = (y))

#define RB_FOREACH_SAFE(x, name, head, y)  \
for ((x) = RB_MIN(name, head);  \
    ((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
    (x) = (y))

#define RB_FOREACH_REVERSE(x, name, head)  \
for ((x) = RB_MAX(name, head);  \
    (x) != NULL;  \
    (x) = name##_RB_PREV(x))

#define RB_FOREACH_REVERSE_FROM(x, name, y)  \
for ((x) = (y);  \
    ((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
    (x) = (y))

```

```
#define RB_FOREACH_REVERSE_SAFE(x, name,
head, y) \
for ((x) = RB_MAX(name, head); \
(x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
(x) = (y))

#endif /* _SYS_TREE_H_ */
pkgname=linux-headers
pkgver=6.14.2 # Follow the latest Linux stable
_kernver=${pkgver%.*}
pkgrel=0
pkgdesc="Linux system headers"
url="https://kernel.org/"
arch="all" # Headers are architecture specific
license="GPL-2.0-only"
makedepends="perl"
options="!check !dbg !strip !tracedeps !archcheck"
source="https://kernel.org/pub/linux/kernel/v6.x/linux-${_kernver}.tar.xz
revert-broken-uapi.patch
0003-remove-inclusion-of-sysinfo.h-in-kernel.h.patch
```

# 1.178 packaging 24.2

## 1.178.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership

of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,



worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

#### END OF TERMS AND CONDITIONS

This software is made available under the terms of *\*either\** of the licenses found in LICENSE.APACHE or LICENSE.BSD. Contributions to this software is made under the terms of *\*both\** these licenses.

Copyright (c) Donald Stufft and individual contributors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# 1.179 ncurses 6.5\_p20250503-r0

## 1.179.1 Available under license :

```
# Contributor: Patrycja Rosa <alpine@ptrcnull.me>
# Maintainer: Patrycja Rosa <alpine@ptrcnull.me>
pkgname=bsd-compat-headers
pkgver=0.7.2
pkgrel=6
pkgdesc="BSD compatibility headers (cdefs, queue, tree)"
url="https://gitlab.alpinelinux.org/alpine/aports"
arch="noarch"
license="BSD-2-Clause AND BSD-3-Clause"
source="
cdefs.h
queue.h
tree.h
"
buildddir="$srcdir"
options="!check" # just headers

package() {
    mkdir -p "$pkgdir"
    install -Dm644 -t "$pkgdir"/usr/include/sys \
        cdefs.h queue.h tree.h
}
sha512sums="
37c8fc73c7aea7b490f7850927e2bb91d12137c9e59e22c084146d515696dbc7973b5de92f4c987ba080dd2502ba8312
7006442c3f019b6447a620c0cae73178 cdefs.h
2f0d5e6e4dc3350285cf17009265dddcbe12431c111868eea39bc8cb038ab7c1f2acacbb21735c4e9d4a1fd106a8fc0f86
11ea33987d4faba37dde5ce6da0750 queue.h
d9ac210d81feb8ad2655bc80fb065d3fe20ae4417b32b4a1711e6738a4870140005c13373b5d1846ef3ce5ae6da45f2da
cef2092881eded0a2e94f6a07752ef3 tree.h
"

# Automatically generated by apkbuild-cpan, template 4
# Contributor: Valery Kartel <valery.kartel@gmail.com>
# Maintainer: Celeste <cielesti@protonmail.com>
maintainer="Celeste <cielesti@protonmail.com>"
pkgname=perl-bsd-resource
pkgver=1.2911
pkgrel=10
#_pkgreal is used by apkbuild-cpan to find modules at MetaCpan
_pkgreal=BSD-Resource
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="https://metacpan.org/release/BSD-Resource/"
arch="all"
license="Artistic-2.0 OR LGPL-2.0"
```

```

depends="perl"
makedepends="perl-dev"
subpackages="$pkgname-doc"
source="https://cpan.metacpan.org/authors/id/J/JH/JHI/BSD-Resource-$pkgver.tar.gz"
builddir="$srcdir/$_pkgreal-$pkgver"

build() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    PERL_MM_USE_DEFAULT=1 perl -I. Makefile.PL \
        INSTALLDIRS=vendor \
        NO_PACKLIST=1 \
        NO_PERLLOCAL=1
    make
}

check() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    make test
}

package() {
    make DESTDIR="$pkgdir" install
}

sha512sums="
d0032d41c7c0468ed1c6d8f57b885f6cb97a5039d754c8cb60b2067daedaf53bd15fb6561a3d0f828df16dfa5417f663b
8065ba65f4fea16dc9262728b3b6b85
    BSD-Resource-1.2911.tar.gz
"

# Contributor: Fabian Affolter <fabian@affolter-engineering.ch>
# Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>
pkgname=py3-flake8-copyright
_pkgname=flake8-copyright
pkgver=0.2.4
pkgrel=3
pkgdesc="Extension for flake8 which checks for copyrights"
options="!check" # No testsuite
url="https://github.com/savoirfairelinux/flake8-copyright"
arch="noarch"
license="MIT"
depends="py3-flake8 py3-setuptools"
makedepends="py3-gpep517 py3-wheel"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/${_pkgname:0:1}/${_pkgname}/${_pkgname}-${pkgver}.tar.gz"
builddir="$srcdir"/$_pkgname-$pkgver

replaces="py-flake8-copyright" # Backwards compatibility
provides="py-flake8-copyright=$pkgver-r$pkgrel" # Backwards compatibility

```

```

build() {
    gpep517 build-wheel \
        --wheel-dir .dist \
        --output-fd 3 3>&1 >&2
}

package() {
    gpep517 install-wheel --destdir "$pkgdir" \
        .dist/*.whl
}

sha512sums="
a6ab47e1bb715618f075c51f398e18180404871b3b0faf1c2d30701d5203db6bc23771bf22ffb5bafcd8ee856b9b64237
b316bd503ae7dceed6ca284ccd5a74d
    flake8-copyright-0.2.4.tar.gz
"

#ifndef _SYS_CDEFS_H_
#define _SYS_CDEFS_H_

#warning usage of non-standard #include <sys/cdefs.h> is deprecated

#undef __P
#undef __PMT

#define __P(args) args
#define __PMT(args) args

#define __CONCAT(x,y) x ## y
#define __STRING(x) #x

#ifdef __cplusplus
# define __BEGIN_DECLS extern "C" {
# define __END_DECLS }
#else
# define __BEGIN_DECLS
# define __END_DECLS
#endif

#if defined(__GNUC__) && !defined(__cplusplus)
# define __THROW __attribute__((__nothrow__))
# define __NTH(fct) __attribute__((__nothrow__)) fct
#else
# define __THROW
# define __NTH(fct) fct
#endif

#endif /* _SYS_CDEFS_H_ */

```

```

# Maintainer: fossdd <fossdd@pwned.life>
pkgname=cargo-license
pkgver=0.6.1
pkgrel=1
pkgdesc="Cargo subcommand to see license of dependencies"
url="https://github.com/onur/cargo-license"
arch="all"
license="MIT"
makedepends="cargo cargo-auditable"
source="$pkgname-$pkgver.tar.gz::https://crates.io/api/v1/crates/cargo-license/$pkgver/download"

prepare() {
    default_prepare

    cargo fetch --target="$CTARGET" --locked
}

build() {
    cargo auditable build --release --frozen
}

check() {
    cargo test --frozen
}

package() {
    install -Dm755 target/release/cargo-license -t "$pkgdir"/usr/bin
}

sha512sums="
36215145e79965be090ae2f5ce6f5eb208465eb67a94c26bed1fe70a131c46c6dc35695e30852759b7366883fbc3caf00
272d796ac634f0a5e1d49232b6b29d5  cargo-license-0.6.1.tar.gz
"

/* $NetBSD: tree.h,v 1.20 2013/09/14 13:20:45 joerg Exp $ */
/* $OpenBSD: tree.h,v 1.13 2011/07/09 00:19:45 pirofti Exp $ */
/* Modified by Void Linux. */
/*
* Copyright 2002 Niels Provos <provos@citi.umich.edu>
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.

```

```

*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE
  AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```

```

#ifndef _SYS_TREE_H_
#define _SYS_TREE_H_

```

```

#ifdef __GNUC__
#define __GNUC_PREREQ__(x, y) \
  ((__GNUC__ == (x) && __GNUC_MINOR__ >= (y)) || \
   (__GNUC__ > (x)))
#else
#define __GNUC_PREREQ__(x, y) 0
#endif

```

```

#if __GNUC_PREREQ__(2, 7) || defined(__lint__)
#define _sys_tree_h_unused __attribute__((__unused__))
#else
#define _sys_tree_h_unused /* delete */
#endif

```

```

/*
* This file defines data structures for different types of trees:
* splay trees and red-black trees.
*
* A splay tree is a self-organizing
  data structure. Every operation
* on the tree causes a splay to happen. The splay moves the requested
* node to the root of the tree and partly rebalances it.
*
* This has the benefit that request locality causes faster lookups as
* the requested nodes move to the top of the tree. On the other hand,
* every lookup causes memory writes.
*
* The Balance Theorem bounds the total access time for m operations
* and n inserts on an initially empty tree as O((m + n)lg n). The
* amortized cost for a sequence of m accesses to a splay tree is O(lg n);
*

```



```

* A red-black tree is a binary search tree with the node color as an
* extra attribute. It fulfills a set of conditions:
* - every search path from the root to a leaf consists of the
*   same number of black nodes,
* - each red node (except for the root) has a black parent,
* - each leaf node is black.
*
* Every operation on a red-black tree is bounded as  $O(\lg n)$ .
* The maximum height of a red-black tree is  $2\lg(n+1)$ .
*/

#define SPLAY_HEAD(name, type) \
struct name { \
    struct type *sph_root; /* root of the tree */ \
}

#define SPLAY_INITIALIZER(root) \
{ NULL }

#define SPLAY_INIT(root) do { \
    (root)->sph_root = NULL; \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ENTRY(type) \
struct { \
    struct type *spe_left; /* left element */ \
    struct type *spe_right; /* right element */ \
}

#define SPLAY_LEFT(elm, field) (elm)->field.spe_left
#define SPLAY_RIGHT(elm, field) (elm)->field.spe_right
#define SPLAY_ROOT(head) (head)->sph_root
#define SPLAY_EMPTY(head) (SPLAY_ROOT(head) == NULL)

/* SPLAY_ROTATE_{LEFT,RIGHT} expect that tmp hold SPLAY_{RIGHT,LEFT} */
#define SPLAY_ROTATE_RIGHT(head, tmp, field) do { \
    SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(tmp, field); \
    SPLAY_RIGHT(tmp, field) = (head)->sph_root; \
    (head)->sph_root = tmp; \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ROTATE_LEFT(head, tmp, field) do { \
    SPLAY_RIGHT((head)->sph_root, \
    field) = SPLAY_LEFT(tmp, field); \
    SPLAY_LEFT(tmp, field) = (head)->sph_root; \
    (head)->sph_root = tmp; \
} while (/*CONSTCOND*/ 0)

```

```

#define SPLAY_LINKLEFT(head, tmp, field) do { \
    SPLAY_LEFT(tmp, field) = (head)->sph_root; \
    tmp = (head)->sph_root; \
    (head)->sph_root = SPLAY_LEFT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKRIGHT(head, tmp, field) do { \
    SPLAY_RIGHT(tmp, field) = (head)->sph_root; \
    tmp = (head)->sph_root; \
    (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ASSEMBLE(head, node, left, right, field) do { \
    SPLAY_RIGHT(left, field) = SPLAY_LEFT((head)->sph_root, field); \
    SPLAY_LEFT(right, field) = SPLAY_RIGHT((head)->sph_root, field); \
    SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(node, field); \
    SPLAY_RIGHT((head)->sph_root, field) = SPLAY_LEFT(node, field); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */

#define
    SPLAY_PROTOTYPE(name, type, field, cmp) \
void name##_SPLAY(struct name *, struct type *); \
void name##_SPLAY_MINMAX(struct name *, int); \
struct type *name##_SPLAY_INSERT(struct name *, struct type *); \
struct type *name##_SPLAY_REMOVE(struct name *, struct type *); \
\
/* Finds the node with the same key as elm */ \
static __inline struct type * \
name##_SPLAY_FIND(struct name *head, struct type *elm) \
{ \
    if (SPLAY_EMPTY(head)) \
        return(NULL); \
    name##_SPLAY(head, elm); \
    if ((cmp)(elm, (head)->sph_root) == 0) \
        return (head->sph_root); \
    return (NULL); \
} \
\
static __inline _sys_tree_h_unused struct type * \
name##_SPLAY_NEXT(struct name *head, struct type *elm) \
{ \
    name##_SPLAY(head, elm); \
    if (SPLAY_RIGHT(elm, field) != NULL) { \
        elm = SPLAY_RIGHT(elm, field); \
        while (SPLAY_LEFT(elm, field) != NULL) { \
            elm = SPLAY_LEFT(elm, field); \

```

```

    } \
}
else \
    elm = NULL; \
return (elm); \
} \
\
static __sys_tree_h_unused __inline struct type * \
name##_SPLAY_MIN_MAX(struct name *head, int val) \
{ \
    name##_SPLAY_MINMAX(head, val); \
    return (SPLAY_ROOT(head)); \
}

/* Main splay operation.
 * Moves node close to the key of elm to top
 */
#define SPLAY_GENERATE(name, type, field, cmp) \
struct type * \
name##_SPLAY_INSERT(struct name *head, struct type *elm) \
{ \
    if (SPLAY_EMPTY(head)) { \
        SPLAY_LEFT(elm, field) = SPLAY_RIGHT(elm, field) = NULL; \
    } else { \
        int __comp; \
        name##_SPLAY(head, elm); \
        __comp = (cmp)(elm, (head)->sph_root); \
        if (__comp < 0) { \
            SPLAY_LEFT(elm, field) = SPLAY_LEFT((head)->sph_root, field); \
            SPLAY_RIGHT(elm, field) = (head)->sph_root; \
            SPLAY_LEFT((head)->sph_root, field) = NULL; \
        } else if (__comp > 0) { \
            SPLAY_RIGHT(elm, \
field) = SPLAY_RIGHT((head)->sph_root, field); \
            SPLAY_LEFT(elm, field) = (head)->sph_root; \
            SPLAY_RIGHT((head)->sph_root, field) = NULL; \
        } else \
            return ((head)->sph_root); \
    } \
    (head)->sph_root = (elm); \
    return (NULL); \
} \
\
struct type * \
name##_SPLAY_REMOVE(struct name *head, struct type *elm) \
{ \
    struct type *__tmp; \
    if (SPLAY_EMPTY(head)) \

```

```

return (NULL); \
name##_SPLAY(head, elm); \
if ((cmp)(elm, (head)->sph_root) == 0) { \
    if (SPLAY_LEFT((head)->sph_root, field) == NULL) { \
        (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field);\
    } else { \
        __tmp = SPLAY_RIGHT((head)->sph_root, field); \
        (head)->sph_root = SPLAY_LEFT((head)->sph_root, field);\
        name##_SPLAY(head, elm); \
        SPLAY_RIGHT((head)->sph_root, field) = __tmp; \
    } \
    return (elm); \
} \
return (NULL); \
} \
\
void \
name##_SPLAY(struct
name *head, struct type *elm) \
{ \
    struct type __node, *__left, *__right, *__tmp; \
    int __comp; \
    \
    SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
    __left = __right = &__node; \
    \
    while ((__comp = (cmp)(elm, (head)->sph_root)) != 0) { \
        if (__comp < 0) { \
            __tmp = SPLAY_LEFT((head)->sph_root, field); \
            if (__tmp == NULL) \
                break; \
            if ((cmp)(elm, __tmp) < 0){ \
                SPLAY_ROTATE_RIGHT(head, __tmp, field); \
                if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
                    break; \
            } \
            SPLAY_LINKLEFT(head, __right, field); \
        } else if (__comp > 0) { \
            __tmp = SPLAY_RIGHT((head)->sph_root, field); \
            if (__tmp == NULL) \
                break; \
            if ((cmp)(elm, __tmp) > 0){ \
                SPLAY_ROTATE_LEFT(head, __tmp, field); \
                if (SPLAY_RIGHT((head)->sph_root, field) == NULL)\
                    break; \
            } \
            SPLAY_LINKRIGHT(head, __left, field); \
        } \
    }

```

```

} \
SPLAY_ASSEMBLE(head,
&__node, __left, __right, field); \
} \
\
/* Splay with either the minimum or the maximum element \
* Used to find minimum or maximum element in tree. \
*/ \
void name##_SPLAY_MINMAX(struct name *head, int __comp) \
{ \
    struct type __node, *__left, *__right, *__tmp; \
    \
    SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL; \
    __left = __right = &__node; \
    \
    while (1) { \
        if (__comp < 0) { \
            __tmp = SPLAY_LEFT((head)->sph_root, field); \
            if (__tmp == NULL) \
                break; \
            if (__comp < 0){ \
                SPLAY_ROTATE_RIGHT(head, __tmp, field); \
                if (SPLAY_LEFT((head)->sph_root, field) == NULL) \
                    break; \
            } \
            SPLAY_LINKLEFT(head, __right, field); \
        } else if (__comp > 0) { \
            __tmp = SPLAY_RIGHT((head)->sph_root, field); \
            if (__tmp == NULL) \
                break; \
            if (__comp > 0) { \
                SPLAY_ROTATE_LEFT(head, __tmp, field); \
                if (SPLAY_RIGHT((head)->sph_root, \
field) == NULL) \
                    break; \
            } \
            SPLAY_LINKRIGHT(head, __left, field); \
        } \
    } \
    SPLAY_ASSEMBLE(head, &__node, __left, __right, field); \
}

```

```

#define SPLAY_NEGINF -1

```

```

#define SPLAY_INF 1

```

```

#define SPLAY_INSERT(name, x, y) name##_SPLAY_INSERT(x, y)

```

```

#define SPLAY_REMOVE(name, x, y) name##_SPLAY_REMOVE(x, y)

```

```

#define SPLAY_FIND(name, x, y) name##_SPLAY_FIND(x, y)

```

```

#define SPLAY_NEXT(name, x, y) name##_SPLAY_NEXT(x, y)
#define SPLAY_MIN(name, x) (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_NEGINF))
#define SPLAY_MAX(name, x) (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_INF))

#define SPLAY_FOREACH(x, name, head) \
    for ((x) = SPLAY_MIN(name, head); \
        (x) != NULL; \
        (x) = SPLAY_NEXT(name, head, x))

/* Macros that define a red-black tree */
#define RB_HEAD(name, type) \
    struct name { \
        struct type *rbh_root; /* root of the tree */ \
    }

#define RB_INITIALIZER(root) \
    { \
        NULL \
    }

#define RB_INIT(root) do { \
    (root)->rbh_root = NULL; \
} while (/*CONSTCOND*/ 0)

#define RB_BLACK 0
#define RB_RED 1
#define RB_ENTRY(type) \
    struct { \
        struct type *rbe_left; /* left element */ \
        struct type *rbe_right; /* right element */ \
        struct type *rbe_parent; /* parent element */ \
        int rbe_color; /* node color */ \
    }

#define RB_LEFT(elm, field) (elm)->field.rbe_left
#define RB_RIGHT(elm, field) (elm)->field.rbe_right
#define RB_PARENT(elm, field) (elm)->field.rbe_parent
#define RB_COLOR(elm, field) (elm)->field.rbe_color
#define RB_ROOT(head) (head)->rbh_root
#define RB_EMPTY(head) (RB_ROOT(head) == NULL)

#define RB_SET(elm, parent, field) do { \
    RB_PARENT(elm, field) = parent; \
    RB_LEFT(elm, field) = RB_RIGHT(elm, field) = NULL; \
    RB_COLOR(elm, field) = RB_RED; \
} while (/*CONSTCOND*/ 0)

```

```

#define RB_SET_BLACKRED(black, red, field) do { \
    RB_COLOR(black, field) = RB_BLACK; \
    RB_COLOR(red, field) \
    = RB_RED; \
} while (/*CONSTCOND*/ 0)

#ifndef RB_AUGMENT
#define RB_AUGMENT(x) do {} while (/*CONSTCOND*/ 0)
#endif

#define RB_ROTATE_LEFT(head, elm, tmp, field) do { \
    (tmp) = RB_RIGHT(elm, field); \
    if ((RB_RIGHT(elm, field) = RB_LEFT(tmp, field)) != NULL) { \
        RB_PARENT(RB_LEFT(tmp, field), field) = (elm); \
    } \
    RB_AUGMENT(elm); \
    if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
        if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
            RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
        else \
            RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
    } else \
        (head)->rbh_root = (tmp); \
    RB_LEFT(tmp, field) = (elm); \
    RB_PARENT(elm, field) = (tmp); \
    RB_AUGMENT(tmp); \
    if ((RB_PARENT(tmp, field))) \
        RB_AUGMENT(RB_PARENT(tmp, field)); \
} while (/*CONSTCOND*/ 0)

#define RB_ROTATE_RIGHT(head, elm, tmp, field) do { \
    (tmp) = RB_LEFT(elm, field); \
    if ((RB_LEFT(elm, field) = RB_RIGHT(tmp, field)) \
        != NULL) { \
        RB_PARENT(RB_RIGHT(tmp, field), field) = (elm); \
    } \
    RB_AUGMENT(elm); \
    if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
        if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
            RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
        else \
            RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
    } else \
        (head)->rbh_root = (tmp); \
    RB_RIGHT(tmp, field) = (elm); \
    RB_PARENT(elm, field) = (tmp); \
    RB_AUGMENT(tmp); \
    if ((RB_PARENT(tmp, field))) \

```

```

    RB_AUGMENT(RB_PARENT(tmp, field)); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */
#define RB_PROTOTYPE(name, type, field, cmp) \
    RB_PROTOTYPE_INTERNAL(name, type, field, cmp,)
#define RB_PROTOTYPE_STATIC(name, type, field, cmp) \
    RB_PROTOTYPE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_PROTOTYPE_INTERNAL(name, type, field, cmp, attr) \
    attr void name##_RB_INSERT_COLOR(struct name *, struct \
    type *); \
    attr void name##_RB_REMOVE_COLOR(struct name *, struct type *, struct type *);\
    attr struct type *name##_RB_REMOVE(struct name *, struct type *); \
    attr struct type *name##_RB_INSERT(struct name *, struct type *); \
    attr struct type *name##_RB_FIND(struct name *, struct type *); \
    attr struct type *name##_RB_NFIND(struct name *, struct type *); \
    attr struct type *name##_RB_NEXT(struct type *); \
    attr struct type *name##_RB_PREV(struct type *); \
    attr struct type *name##_RB_MINMAX(struct name *, int); \
    \

/* Main rb operation.
* Moves node close to the key of elm to top
*/
#define RB_GENERATE(name, type, field, cmp) \
    RB_GENERATE_INTERNAL(name, type, field, cmp,)
#define RB_GENERATE_STATIC(name, type, field, cmp) \
    RB_GENERATE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_GENERATE_INTERNAL(name, type, field, cmp, attr) \
    attr void \
    name##_RB_INSERT_COLOR(struct name *head, struct type *elm) \
    { \
        struct type \
        *parent, *gparent, *tmp; \
        while ((parent = RB_PARENT(elm, field)) != NULL && \
            RB_COLOR(parent, field) == RB_RED) { \
            gparent = RB_PARENT(parent, field); \
            if (parent == RB_LEFT(gparent, field)) { \
                tmp = RB_RIGHT(gparent, field); \
                if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
                    RB_COLOR(tmp, field) = RB_BLACK; \
                    RB_SET_BLACKRED(parent, gparent, field);\
                    elm = gparent; \
                    continue; \
                } \
            } \
            if (RB_RIGHT(parent, field) == elm) { \
                RB_ROTATE_LEFT(head, parent, tmp, field);\
                tmp = parent; \

```



```

    parent = elm; \
    elm = tmp; \
} \
RB_SET_BLACKRED(parent, gparent, field); \
RB_ROTATE_RIGHT(head, gparent, tmp, field); \
} else { \
    tmp = RB_LEFT(gparent, field); \
    if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
        RB_COLOR(tmp, field) = RB_BLACK; \
        RB_SET_BLACKRED(parent, gparent, field); \
        elm = gparent; \
        continue; \
    } \
    if (RB_LEFT(parent, field) == elm)
{ \
    RB_ROTATE_RIGHT(head, parent, tmp, field); \
    tmp = parent; \
    parent = elm; \
    elm = tmp; \
} \
    RB_SET_BLACKRED(parent, gparent, field); \
    RB_ROTATE_LEFT(head, gparent, tmp, field); \
} \
} \
RB_COLOR(head->rbh_root, field) = RB_BLACK; \
} \
\
attr void \
name##_RB_REMOVE_COLOR(struct name *head, struct type *parent, struct type *elm) \
{ \
    struct type *tmp; \
    while ((elm == NULL || RB_COLOR(elm, field) == RB_BLACK) && \
        elm != RB_ROOT(head)) { \
        if (RB_LEFT(parent, field) == elm) { \
            tmp = RB_RIGHT(parent, field); \
            if (RB_COLOR(tmp, field) == RB_RED) { \
                RB_SET_BLACKRED(tmp, parent, field); \
                RB_ROTATE_LEFT(head, parent, tmp, field); \
                tmp = RB_RIGHT(parent, field); \
            } \
            if ((RB_LEFT(tmp, field) == NULL || \
                RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) && \
                (RB_RIGHT(tmp, field) == NULL || \
                RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK)) { \
                RB_COLOR(tmp, field) = RB_RED; \
                elm = parent; \
                parent = RB_PARENT(elm, field); \

```

```

} else { \
if (RB_RIGHT(tmp, field) == NULL || \
    RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK) {\
    struct type *oleft; \
    if ((oleft = RB_LEFT(tmp, field)) \
        != NULL) \
        RB_COLOR(oleft, field) = RB_BLACK;\
    RB_COLOR(tmp, field) = RB_RED; \
    RB_ROTATE_RIGHT(head, tmp, oleft, field);\
    tmp = RB_RIGHT(parent, field); \
} \
RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
RB_COLOR(parent, field) = RB_BLACK; \
if (RB_RIGHT(tmp, field)) \
    RB_COLOR(RB_RIGHT(tmp, field), field) = RB_BLACK;\
RB_ROTATE_LEFT(head, parent, tmp, field);\
elm = RB_ROOT(head); \
break; \
} \
} else { \
    tmp = RB_LEFT(parent, field); \
    if (RB_COLOR(tmp, field) == RB_RED) { \
        RB_SET_BLACKRED(tmp, parent, field); \
        RB_ROTATE_RIGHT(head, parent, tmp, field);\
        tmp
= RB_LEFT(parent, field); \
    } \
    if ((RB_LEFT(tmp, field) == NULL || \
        RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) &&\
        (RB_RIGHT(tmp, field) == NULL || \
        RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK)) {\
        RB_COLOR(tmp, field) = RB_RED; \
        elm = parent; \
        parent = RB_PARENT(elm, field); \
    } else { \
        if (RB_LEFT(tmp, field) == NULL || \
            RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) {\
            struct type *oright; \
            if ((oright = RB_RIGHT(tmp, field)) \
                != NULL) \
                RB_COLOR(oright, field) = RB_BLACK;\
            RB_COLOR(tmp, field) = RB_RED; \
            RB_ROTATE_LEFT(head, tmp, oright, field);\
            tmp = RB_LEFT(parent, field); \
        } \
        RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
        RB_COLOR(parent, field) = RB_BLACK; \
        if (RB_LEFT(tmp, field)) \

```

```

    RB_COLOR(RB_LEFT(tmp, field), field) = RB_BLACK;\
    RB_ROTATE_RIGHT(head, parent, tmp, field);\
    elm = RB_ROOT(head); \
    break; \
} \
} \
} \
if
(elm) \
    RB_COLOR(elm, field) = RB_BLACK; \
} \
\
attr struct type * \
name##_RB_REMOVE(struct name *head, struct type *elm) \
{ \
    struct type *child, *parent, *old = elm; \
    int color; \
    if (RB_LEFT(elm, field) == NULL) \
        child = RB_RIGHT(elm, field); \
    else if (RB_RIGHT(elm, field) == NULL) \
        child = RB_LEFT(elm, field); \
    else { \
        struct type *left; \
        elm = RB_RIGHT(elm, field); \
        while ((left = RB_LEFT(elm, field)) != NULL) \
            elm = left; \
        child = RB_RIGHT(elm, field); \
        parent = RB_PARENT(elm, field); \
        color = RB_COLOR(elm, field); \
        if (child) \
            RB_PARENT(child, field) = parent; \
        if (parent) { \
            if (RB_LEFT(parent, field) == elm) \
                RB_LEFT(parent, field) = child; \
            else \
                RB_RIGHT(parent, field) = child; \
            RB_AUGMENT(parent); \
        } else \
            RB_ROOT(head)
= child; \
    if (RB_PARENT(elm, field) == old) \
        parent = elm; \
    (elm)->field = (old)->field; \
    if (RB_PARENT(old, field)) { \
        if (RB_LEFT(RB_PARENT(old, field), field) == old)\
            RB_LEFT(RB_PARENT(old, field), field) = elm;\
        else \
            RB_RIGHT(RB_PARENT(old, field), field) = elm;\
    }

```

```

    RB_AUGMENT(RB_PARENT(old, field)); \
} else \
    RB_ROOT(head) = elm; \
    RB_PARENT(RB_LEFT(old, field), field) = elm; \
    if (RB_RIGHT(old, field)) \
        RB_PARENT(RB_RIGHT(old, field), field) = elm; \
    if (parent) { \
        left = parent; \
        do { \
            RB_AUGMENT(left); \
        } while ((left = RB_PARENT(left, field)) != NULL); \
    } \
    goto color; \
} \
parent = RB_PARENT(elm, field); \
color = RB_COLOR(elm, field); \
if (child) \
    RB_PARENT(child, field) = parent; \
if (parent) { \
    if (RB_LEFT(parent, field) == elm) \
        RB_LEFT(parent, field) = child; \
    else \
        RB_RIGHT(parent,
field) = child; \
    RB_AUGMENT(parent); \
} else \
    RB_ROOT(head) = child; \
color: \
if (color == RB_BLACK) \
    name##_RB_REMOVE_COLOR(head, parent, child); \
return (old); \
} \
\
/* Inserts a node into the RB tree */ \
attr struct type * \
name##_RB_INSERT(struct name *head, struct type *elm) \
{ \
    struct type *tmp; \
    struct type *parent = NULL; \
    int comp = 0; \
    tmp = RB_ROOT(head); \
    while (tmp) { \
        parent = tmp; \
        comp = (cmp)(elm, parent); \
        if (comp < 0) \
            tmp = RB_LEFT(tmp, field); \
        else if (comp > 0) \
            tmp = RB_RIGHT(tmp, field); \
    }

```

```

else    \
    return (tmp);    \
}    \
RB_SET(elm, parent, field);    \
if (parent != NULL) {    \
    if (comp < 0)    \
        RB_LEFT(parent, field) = elm;    \
    else    \
        RB_RIGHT(parent, field) = elm;    \
    RB_AUGMENT(parent);    \
} else    \
    RB_ROOT(head) = elm;    \
name##_RB_INSERT_COLOR(head,
elm);    \
return (NULL);    \
}    \
    \
/* Finds the node with the same key as elm */    \
attr struct type *    \
name##_RB_FIND(struct name *head, struct type *elm)    \
{    \
    struct type *tmp = RB_ROOT(head);    \
    int comp;    \
    while (tmp) {    \
        comp = cmp(elm, tmp);    \
        if (comp < 0)    \
            tmp = RB_LEFT(tmp, field);    \
        else if (comp > 0)    \
            tmp = RB_RIGHT(tmp, field);    \
        else    \
            return (tmp);    \
    }    \
    return (NULL);    \
}    \
    \
/* Finds the first node greater than or equal to the search key */    \
attr struct type *    \
name##_RB_NFIND(struct name *head, struct type *elm)    \
{    \
    struct type *tmp = RB_ROOT(head);    \
    struct type *res = NULL;    \
    int comp;    \
    while (tmp) {    \
        comp = cmp(elm, tmp);    \
        if (comp < 0) {    \
            res = tmp;    \
            tmp = RB_LEFT(tmp, field);    \
        }    \
    }    \
}    \

```

```

else if (comp > 0) \
    tmp =
RB_RIGHT(tmp, field); \
else \
    return (tmp); \
} \
return (res); \
} \
\
/* ARGSUSED */ \
attr struct type * \
name##_RB_NEXT(struct type *elm) \
{ \
    if (RB_RIGHT(elm, field)) { \
        elm = RB_RIGHT(elm, field); \
        while (RB_LEFT(elm, field)) \
            elm = RB_LEFT(elm, field); \
    } else { \
        if (RB_PARENT(elm, field) && \
            (elm == RB_LEFT(RB_PARENT(elm, field), field))) \
            elm = RB_PARENT(elm, field); \
        else { \
            while (RB_PARENT(elm, field) && \
                (elm == RB_RIGHT(RB_PARENT(elm, field), field))) \
                elm = RB_PARENT(elm, field); \
        } \
    } \
    return (elm); \
} \
\
/* ARGSUSED */ \
attr struct type * \
name##_RB_PREV(struct type *elm) \
{ \
    if (RB_LEFT(elm, field)) { \
        elm = RB_LEFT(elm, field); \
        while (RB_RIGHT(elm, field)) \
            elm = RB_RIGHT(elm, field); \
    } \
    else { \
        if (RB_PARENT(elm, field) && \
            (elm == RB_RIGHT(RB_PARENT(elm, field), field))) \
            elm = RB_PARENT(elm, field); \
        else { \
            while (RB_PARENT(elm, field) && \
                (elm == RB_LEFT(RB_PARENT(elm, field), field))) \
                elm = RB_PARENT(elm, field); \
        } \
    } \
}

```

```

    elm = RB_PARENT(elm, field); \
} \
} \
return (elm); \
} \
\
attr struct type * \
name##_RB_MINMAX(struct name *head, int val) \
{ \
    struct type *tmp = RB_ROOT(head); \
    struct type *parent = NULL; \
    while (tmp) { \
        parent = tmp; \
        if (val < 0) \
            tmp = RB_LEFT(tmp, field); \
        else \
            tmp = RB_RIGHT(tmp, field); \
    } \
    return (parent); \
}

#define RB_NEGINF -1
#define RB_INF 1

#define RB_INSERT(name, x, y) name##_RB_INSERT(x, y)
#define RB_REMOVE(name, x, y) name##_RB_REMOVE(x, y)
#define RB_FIND(name, x, y) name##_RB_FIND(x, y)
#define RB_NFIND(name, x, y) name##_RB_NFIND(x, \
y)
#define RB_NEXT(name, x, y) name##_RB_NEXT(y)
#define RB_PREV(name, x, y) name##_RB_PREV(y)
#define RB_MIN(name, x) name##_RB_MINMAX(x, RB_NEGINF)
#define RB_MAX(name, x) name##_RB_MINMAX(x, RB_INF)

#define RB_FOREACH(x, name, head) \
for ((x) = RB_MIN(name, head); \
(x) != NULL; \
(x) = name##_RB_NEXT(x))

#define RB_FOREACH_FROM(x, name, y) \
for ((x) = (y); \
((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
(x) = (y))

#define RB_FOREACH_SAFE(x, name, head, y) \
for ((x) = RB_MIN(name, head); \
((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
(x) = (y))

```

```

#define RB_FOREACH_REVERSE(x, name, head) \
for ((x) = RB_MAX(name, head); \
(x) != NULL; \
(x) = name##_RB_PREV(x))

#define RB_FOREACH_REVERSE_FROM(x, name, y) \
for ((x) = (y); \
((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
(x) = (y))

#define RB_FOREACH_REVERSE_SAFE(x, name, \
head, y) \
for ((x) = RB_MAX(name, head); \
((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
(x) = (y))

#endif /* _SYS_TREE_H_ */
MIT
# MIT License

```

Copyright 2019-2021 Caleb Porzio and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT

OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2023 Leaf Corcoran

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights



to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```
# Maintainer: Natanael Copa <ncopa@alpinelinux.org>
```

```
pkgname=libart-lgpl
```

```
pkgver=2.3.21
```

```
pkgrel=8
```

```
pkgdesc="A library for high-performance 2D graphics"
```

```
url="https://www.levien.com/libart/"
```

```
arch="all"
```

```
options="!check" # No test suite.
```

```
license="LGPL-2.0-or-later"
```

```
subpackages="$pkgname-dev"
```

```
source="https://download.gnome.org/sources/libart-lgpl/2.3/libart-lgpl-$pkgver.tar.bz2"
```

```
"
```

```
builddir="$srcdir"/libart-lgpl-$pkgver
```

```
prepare() {  
    update_config_sub  
    default_prepare  
}
```

```
build() {  
    ./configure \  
    --build=$CBUILD \  
    --host=$CHOST \  
    --prefix=/usr  
    make  
}
```

```
package() {  
    make DESTDIR="$pkgdir" install  
}
```

```
sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7"
```

6654309e001aee9bace727b5b4c8589405256a3c020 libart\_lgpl-2.3.21.tar.bz2"

mini\_sendmail - accept email on behalf of real sendmail

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.180 truststore 0.10.1

### 1.180.1 Available under license :

MIT

The MIT License (MIT)

Copyright (c) 2022 Seth Michael Larson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT

OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.181 c-ares 1.34.5-r0

## 1.181.1 Available under license :

/\* MIT License

\*

\* Copyright (c) John Schember

\*

\* Permission is hereby granted, free of charge, to any person obtaining a copy

\* of this software and associated documentation files (the "Software"), to deal

\* in the Software without restriction, including without limitation the rights

\* to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

\* copies of the Software, and to permit persons to whom the Software is

\* furnished to do so, subject to the following conditions:

\*

\* The above copyright notice and this permission notice (including the next

\* paragraph) shall be included in all copies or substantial portions of the

\* Software.

\*

\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

\* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

\* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

\* AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

\* LIABILITY, WHETHER

IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

\* OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE

\* SOFTWARE.

\*

\* SPDX-License-Identifier: MIT

\*/

MIT License

Copyright (c) 1998 Massachusetts Institute of Technology

Copyright (c) 2007 - 2023 Daniel Stenberg with many contributors, see AUTHORS file.

Permission is hereby granted, free of charge, to any person obtaining a copy of

this software and associated documentation files (the "Software"), to deal in

the Software without restriction, including without limitation the rights to

use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of

the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next paragraph) shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.182 socat 1.8.0.3-r1

### 1.182.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):

\*

No license file was found, but licenses were detected in source scan.

diff -r -N -U 3 --no-dereference socat-1.8.0.2/CHANGES socat-1.8.0.3/CHANGES

Found in path(s):

\* /socat-1-8-0-3-patch-gz/socat-1.8.0.3.patch

## 1.183 clang18 18.1.8-r5

### 1.183.1 Available under license :

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition,

"submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each

Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]



Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions  
of this Software are embedded into an Object form of such source code, you  
may redistribute such embedded portions in such Object form without complying  
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled  
forms of this Software with  
software that is licensed under the GPLv2 ("Combined Software") and if a  
court of competent jurisdiction determines that the patent provision (Section  
3), the indemnity provision (Section 9) or other Section of the License  
conflicts with the conditions of the GPLv2, you may retroactively and  
prospectively choose to deem waived or otherwise exclude such Section(s) of  
the License, but only in their entirety and only with respect to the Combined  
Software.

=====  
Software from third parties included in the LLVM Project:  
=====

The LLVM Project contains third party software which is under different license  
terms. All such code will be identified clearly using at least one of two  
mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or  
`LICENSE` file at the top containing the specific  
license and restrictions  
which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every  
file.

=====  
Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):  
=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2007-2019 University of Illinois at Urbana-Champaign.  
All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

<http://llvm.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do

so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- \* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

## APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

#### ---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions  
of this Software are embedded into an Object form of such source code, you  
may redistribute such embedded portions in such Object form without complying  
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled  
forms of this Software with  
software that is licensed under the GPLv2 ("Combined Software") and if a  
court of competent jurisdiction determines that the patent provision (Section  
3), the indemnity provision (Section 9) or other Section of the License  
conflicts with the conditions of the GPLv2, you may retroactively and  
prospectively choose to deem waived or otherwise exclude such Section(s) of  
the License, but only in their entirety and only with respect to the Combined  
Software.

=====  
Software from third parties included in the LLVM Project:  
=====

The LLVM Project contains third party software which is under different license  
terms. All such code will be identified clearly using at least one of two  
mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or  
`LICENSE` file at the top containing the specific  
license and restrictions  
which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every  
file.

=====  
Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):

=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2007-2018 University of Illinois at Urbana-Champaign.  
All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

<http://llvm.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.

- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.

- \* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

## 1.184 bird 3.1.4-r0

### 1.184.1 Available under license :

This is 'linuxdoc-tools', a series of tools to implement the Linux Documentation Project HOWTO and book styles in SGML.

This copy was modified in order to work with birddoc DTD. It is



based on version 0.9.73-2 from Debian 10.

-----  
The linuxdoc-tools license  
-----

linuxdoc-tools is derived from linuxdoc-SGML, originally written by Matt Welsh and later maintained by Cees de Groot. Linuxdoc-SGML is based on James Clark's sgmls parser, and the QWERTZ DTD by Tom Gordon. Magnus Alvestad provided the current HTML support. For the rest of linuxdoc-SGML,

Copyright (C) 1994-1996 Matt Welsh <mdw@cs.cornell.edu>

Copyright (C) 1996-1998 Cees de Groot <cg@pobox.com>

Original Linuxdoc-SGML itself does not have any limitations. Everything not having explicit additional conditions can be freely used, modified, and redistributed, under the usual fair use clauses:

- \* No warranty. Use at your own risk.
- \* Do not pretend to have written what you did not (Preserve credits and Copyright notices of the different elements if present).

Since then, lots of smaller and bigger changes resulted in a rename to SGML-Tools (and then to SGMLtools, the hyphen caused confusion) to indicate that it wasn't just for Linux anymore. See files CHANGES.old-v1 and CONTRIBUTORS.old-v1 for changelog and list of contributors to old linuxdoc-sgml and sgmltools-v1.

When sgml-tools dropped support for the linuxdoc DTD, Taketoshi Sano <sano@debian.org> forked the code to linuxdoc-tools. See README file.

Changes after the fork

Copyright (C) 1999-2002 Taketoshi Sano <sano@debian.org>

Copyright (C) 2000 Juan Jose Amor

Copyright (C) 2007-2018 Agustin Martin Domingo <agmartin@debian.org>

Unless conflicting with other licenses, changes by Agustin Martin Domingo are free software: you can redistribute and/or modify them under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. Otherwise they honour previous license.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>. In Debian systems you can find a copy under /usr/share/common-licenses.

-----  
The entity-map license  
-----

Copyright (C) 1997 Ken MacLeod

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL KEN MACLEOD BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of Ken MacLeod shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from Ken MacLeod.

The patch is provided under the terms of the GNU General Public License, either version 2, or any later version.

/\*

\* Configuration for FreeBSD based systems with netlink support

\*

\* (c) 2022 Alexander Chernikov <[melifaro@FreeBSD.org](mailto:melifaro@FreeBSD.org)>

\*

\* Can be freely distributed and used under the terms of the GNU GPL.

\*/

```
#define CONFIG_AUTO_ROUTES
#define CONFIG_SELF_CONSCIOUS
#define CONFIG_MULTIPLE_TABLES
```

```

#define CONFIG_SINGLE_ROUTE
#define CONFIG_ASSUME_ONLINK

#define CONFIG_SKIP_MC_BIND
#define CONFIG_NO_IFACE_BIND
#define CONFIG_USE_HDRINCL

#define CONFIG_INCLUDE_SYSIO_H "sysdep/bsd/sysio.h"
#define CONFIG_INCLUDE_KRTSYS_H "sysdep/linux/krt-sys.h"
#define CONFIG_INCLUDE_NLSYS_H "sysdep/bsd-netlink/netlink-sys.h"

/*
Link: sysdep/unix
Link: sysdep/bsd-netlink
*/
/*
* Configuration for *BSD based systems (tested on FreeBSD and NetBSD)
*
* (c) 2004 Ondrej Filip <feela@network.cz>
*
* Can be freely distributed and used under the terms of the GNU GPL.
*/

#define CONFIG_AUTO_ROUTES
#define CONFIG_SELF_CONSCIOUS
#define CONFIG_MULTIPLE_TABLES
#define CONFIG_SINGLE_ROUTE
#define CONFIG_ASSUME_ONLINK

#define CONFIG_SKIP_MC_BIND
#define CONFIG_NO_IFACE_BIND
#define CONFIG_USE_HDRINCL

#define CONFIG_INCLUDE_SYSIO_H "sysdep/bsd/sysio.h"
#define CONFIG_INCLUDE_KRTSYS_H "sysdep/bsd/krt-sys.h"

/*
Link: sysdep/unix
Link: sysdep/bsd
*/
/*
* Netlink FreeBSD-specific functions
*
* (c) 2022 Alexander Chernikov <melifaro@FreeBSD.org>
*
* Can be freely distributed and used under the terms of the GNU GPL.
*/

```

```

#ifndef _BIRD_NETLINK_SYS_H_
#define _BIRD_NETLINK_SYS_H_

#include <netlink/netlink.h>
#include <netlink/netlink_route.h>

#ifndef AF_MPLS
#define AF_MPLS 39
#endif

#ifndef SO_RCVBUFFORCE
#define SO_RCVBUFFORCE SO_RCVBUF
#endif

static inline int
netlink_error_to_os(int error)
{
    return (error);
}

#endif
Format: https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Upstream-Name: bird
Upstream-Contact: bird-users@network.cz
Source: https://bird.network.cz

Files: *
Copyright: 1998-2008 Martin Mare
          1998-2019 Ondej Filip
          1998-2000 Pavel Machek
          2008-2019 Ondej Zajek
          2015-2019 Maria Matjka
License: GPL-2+

Files: lib/heap.h
Copyright: 2001 Martin Mare <mj@ucw.cz>
          2005 Tom Valla <tom@ucw.cz>
License: LGPL-2+

Files: lib/md5.c
Copyright: 1993 Colin Plumb
License: public-domain

Files: lib/printf.c
Copyright: 1991-1992 Lars Wirzenius
          1991-1992 Linus Torvalds
          1998-2000 Martin Mare
License: GPL-2

```

Files: lib/sha\*

Copyright: 1998-2009 Free Software Foundation, Inc.

2015-2016 Pavel Tvrdek

2015-2016 Ondrej Zatek

License: GPL-2+

Files: proto/babel/\*

Copyright: 2015-2016 Toke Hild-Jrgensen

2016-2018 Ondrej Zatek

License: GPL-2+

Files: proto/rpki/\*

Copyright: 2015-2016 Pavel Tvrdek

2016-2018 Ondrej Zatek

License: GPL-2+

Files: debian/\*

Copyright:

2010-2013 Ondrej Sur <ondrej@debian.org>

License: GPL-2+

License: GPL-2+

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

.

On Debian systems, the complete text of the GNU General Public License version 2 can be found in `/usr/share/common-licenses/GPL-2'.

License: GPL-2

This package is free software; you can redistribute it and/or modify

it under the terms of the GNU General Public License version 2 as published by the Free Software Foundation.

.

This package is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of

MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this package; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

On Debian systems, the complete text of the GNU General Public License version 2 can be found in `/usr/share/common-licenses/GPL-2'.

License: LGPL-2+

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

On Debian systems, the complete text of the GNU Lesser General Public License version 2 can be found in `/usr/share/common-licenses/LGPL-2'.

License: public-domain

This code is in the public domain; do with it what you wish.

## 1.185 llvm18 18.1.8-r5

### 1.185.1 Available under license :

; NOTE: Assertions have been autogenerated by utils/update\_llc\_test\_checks.py

; RUN: llc < %s -mtriple=aarch64-- | FileCheck %s

; A shuffle mask with all undef elements is always legal.

```
define <4 x i32> @PR41535(<2 x i32> %p1, <2 x i32> %p2) {
; CHECK-LABEL: PR41535:
; CHECK:      // %bb.0:
; CHECK-NEXT:  ext v0.8b, v0.8b, v1.8b, #4
; CHECK-NEXT:  mov v0.d[1], v0.d[0]
; CHECK-NEXT:  ret
%cat1 = shufflevector <2 x i32> %p1, <2 x i32> undef, <4 x i32> <i32 undef, i32 1, i32 undef, i32 undef>
%cat2 = shufflevector <2 x i32> %p2, <2 x i32> undef, <4 x i32> <i32 0, i32 undef, i32 undef, i32 undef>
%r = shufflevector <4 x i32> %cat1, <4 x i32> %cat2, <4 x i32> <i32 undef, i32 undef, i32 1, i32 4>
ret <4 x i32> %r
}
```

```
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=amdgc -mcpu=fiji < %s | FileCheck -enable-var-scope --check-prefixes=GCN,GCN-
SAFE,VI,VI-SAFE %s
; RUN: llc -enable-no-signed-zeros-fp-math -mtriple=amdgc -mcpu=fiji < %s | FileCheck -enable-var-scope --
check-prefixes=GCN,GCN-NSZ,VI,VI-NSZ %s
```

```
; -----
; rcp tests
; -----
```

```
define half @v_fneg_rcp_f16(half %a) #0 {
; GCN-LABEL: v_fneg_rcp_f16:
; GCN:      ; %bb.0:
; GCN-NEXT:  s_waitcnt vmcnt(0) expcnt(0) lgkmcnt(0)
; GCN-NEXT:  v_rcp_f16_e64 v0, -v0
; GCN-NEXT:  s_setpc_b64 s[30:31]
%rcp = call half @llvm.amdgc.rcp.f16(half %a)
%fneg = fneg half %rcp
ret half %fneg
}
```

```
define half @v_fneg_rcp_fneg_f16(half %a) #0 {
; GCN-LABEL: v_fneg_rcp_fneg_f16:
; GCN:      ; %bb.0:
; GCN-NEXT:  s_waitcnt vmcnt(0) expcnt(0) lgkmcnt(0)
; GCN-NEXT:
    v_rcp_f16_e32 v0, v0
; GCN-NEXT:  s_setpc_b64 s[30:31]
%fneg.a = fneg half %a
%rcp = call half @llvm.amdgc.rcp.f16(half %fneg.a)
%fneg = fneg half %rcp
ret half %fneg
}
```

```
define { half, half } @v_fneg_rcp_store_use_fneg_f16(half %a) #0 {
; GCN-LABEL: v_fneg_rcp_store_use_fneg_f16:
; GCN:      ; %bb.0:
; GCN-NEXT:  s_waitcnt vmcnt(0) expcnt(0) lgkmcnt(0)
; GCN-NEXT:  v_rcp_f16_e32 v2, v0
; GCN-NEXT:  v_xor_b32_e32 v1, 0x8000, v0
; GCN-NEXT:  v_mov_b32_e32 v0, v2
; GCN-NEXT:  s_setpc_b64 s[30:31]
%fneg.a = fneg half %a
%rcp = call half @llvm.amdgc.rcp.f16(half %fneg.a)
%fneg = fneg half %rcp
%insert.0 = insertvalue { half, half } poison, half %fneg, 0
%insert.1 = insertvalue { half, half } %insert.0, half %fneg.a, 1
ret { half, half } %insert.1
}
```

```

}

define { half, half } @v_fneg_rcp_multi_use_fneg_f16(half %a, half %c) #0 {
; GCN-LABEL: v_fneg_rcp_multi_use_fneg_f16:
; GCN:      ; %bb.0:
; GCN-NEXT:  s_waitcnt vmcnt(0) expcnt(0) lgkmcnt(0)
; GCN-NEXT:
    v_rcp_f16_e32 v2, v0
; GCN-NEXT:  v_mul_f16_e64 v1, -v0, v1
; GCN-NEXT:  v_mov_b32_e32 v0, v2
; GCN-NEXT:  s_setpc_b64 s[30:31]
    %fneg.a = fneg half %a
    %rcp = call half @llvm.amdgcn.rcp.f16(half %fneg.a)
    %fneg = fneg half %rcp
    %use1 = fmul half %fneg.a, %c
    %insert.0 = insertvalue { half, half } poison, half %fneg, 0
    %insert.1 = insertvalue { half, half } %insert.0, half %use1, 1
    ret { half, half } %insert.1
}

```

```

; -----
; sin tests
; -----

```

```

define half @v_fneg_amdgcn_sin_f16(half %a) #0 {
; GCN-LABEL: v_fneg_amdgcn_sin_f16:
; GCN:      ; %bb.0:
; GCN-NEXT:  s_waitcnt vmcnt(0) expcnt(0) lgkmcnt(0)
; GCN-NEXT:  v_sin_f16_e64 v0, -v0
; GCN-NEXT:  s_setpc_b64 s[30:31]
    %sin = call half @llvm.amdgcn.sin.f16(half %a)
    %fneg = fneg half %sin
    ret half %fneg
}

```

```

; -----
;
; vinterp tests
; -----

```

```

define { float, float } @v_fneg_interp_p1_f16(float %a, float %b) #0 {
; GCN-LABEL: v_fneg_interp_p1_f16:
; GCN:      ; %bb.0:
; GCN-NEXT:  s_waitcnt vmcnt(0) expcnt(0) lgkmcnt(0)
; GCN-NEXT:  v_mul_f32_e64 v1, v0, -v1
; GCN-NEXT:  s_mov_b32 m0, 0
; GCN-NEXT:  s_setreg_imm32_b32 hwreg(HW_REG_MODE, 2, 2), 3
; GCN-NEXT:  v_interp_p1ll_f16 v0, v1, attr0.x

```



```

; GCN-NEXT:  v_interp_p1l_f16 v1, v1, attr0.y
; GCN-NEXT:  s_setpc_b64 s[30:31]
%mul = fmul float %a, %b
%fneg = fneg float %mul
%intrp0 = call float @llvm.amdgcn.interp.p1.f16(float %fneg, i32 0, i32 0, i1 false, i32 0)
%intrp1 = call float @llvm.amdgcn.interp.p1.f16(float %fneg, i32 1, i32 0, i1 false, i32 0)
%insert.0 = insertvalue { float, float } poison, float %intrp0, 0
%insert.1 = insertvalue { float, float } %insert.0, float %intrp1, 1
ret { float, float } %insert.1
}

define
{ half, half } @v_fneg_interp_p2_f16(float %a, float %b) #0 {
; GCN-LABEL: v_fneg_interp_p2_f16:
; GCN:      ; %bb.0:
; GCN-NEXT:  s_waitcnt vmcnt(0) expcnt(0) lgkmcnt(0)
; GCN-NEXT:  v_mul_f32_e64 v1, v0, -v1
; GCN-NEXT:  v_mov_b32_e32 v2, 4.0
; GCN-NEXT:  s_mov_b32 m0, 0
; GCN-NEXT:  s_setreg_imm32_b32 hwreg(HW_REG_MODE, 2, 2), 3
; GCN-NEXT:  v_interp_p2_f16 v0, v1, attr0.x, v2
; GCN-NEXT:  v_interp_p2_f16 v1, v1, attr0.y, v2
; GCN-NEXT:  s_setpc_b64 s[30:31]
%mul = fmul float %a, %b
%fneg = fneg float %mul
%intrp0 = call half @llvm.amdgcn.interp.p2.f16(float 4.0, float %fneg, i32 0, i32 0, i1 false, i32 0)
%intrp1 = call half @llvm.amdgcn.interp.p2.f16(float 4.0, float %fneg, i32 1, i32 0, i1 false, i32 0)
%insert.0 = insertvalue { half, half } poison, half %intrp0, 0
%insert.1 = insertvalue { half, half } %insert.0, half %intrp1, 1
ret { half, half } %insert.1
}

; -----
;
; arithmetic.fence tests
; -----

; FIXME: Legalization/promote is broken
define half @v_fneg_arithmetic_fence_f16(half %a) #0 {
; GCN-LABEL: v_fneg_arithmetic_fence_f16:
; GCN:      ; %bb.0:
; GCN-NEXT:  ; ARITH_FENCE
; GCN-NEXT:  s_waitcnt vmcnt(0) expcnt(0) lgkmcnt(0)
; GCN-NEXT:  v_xor_b32_e32 v0, 0x8000, v0
; GCN-NEXT:  s_setpc_b64 s[30:31]
%fence = call half @llvm.arithmetic.fence.f16(half %a)
%fneg = fneg half %fence
ret half %fneg

```

```

}

define half @v_fneg_arithmetic_fence_fmul_f16(half %a, half %b) #0 {
; GCN-LABEL: v_fneg_arithmetic_fence_fmul_f16:
; GCN:      ; %bb.0:
; GCN-NEXT:  s_waitcnt vmcnt(0) expcnt(0) lgkmcnt(0)
; GCN-NEXT:  v_mul_f16_e32 v0, v0, v1
; GCN-NEXT:  ;ARITH_FENCE
; GCN-NEXT:  v_xor_b32_e32 v0, 0x8000, v0
; GCN-NEXT:  s_setpc_b64 s[30:31]
%mul = fmul half %a, %b
%fence = call half @llvm.arithmetic.fence.f16(half %mul)
%fneg = fneg half %fence
ret half
%fneg
}

declare half @llvm.amdgcn.rcp.f16(half) #1
declare half @llvm.amdgcn.sin.f16(half) #1
declare half @llvm.arithmetic.fence.f16(half) #1
declare float @llvm.amdgcn.interp.p1.f16(float, i32, i32, i1, i32) #0
declare half @llvm.amdgcn.interp.p2.f16(float, float, i32, i32, i1, i32) #0

attributes #0 = { nounwind "denormal-fp-math-f32"="preserve-sign,preserve-sign" }
attributes #1 = { nounwind readnone }
attributes #2 = { nounwind "unsafe-fp-math"="true" }
attributes #3 = { nounwind "no-signed-zeros-fp-math"="true" }
attributes #4 = { nounwind "amdgpu-ieee"="false" "denormal-fp-math-f32"="preserve-sign,preserve-sign" }
;; NOTE: These prefixes are unused and the list is autogenerated. Do not add tests below this line:
; GCN-NSZ: {{.*}}
; GCN-SAFE: {{.*}}
; VI: {{.*}}
; VI-NSZ: {{.*}}
; VI-SAFE: {{.*}}
; NOTE: Assertions have been autogenerated by utils/update_test_checks.py UTC_ARGS: --function-signature --
check-attributes --check-globals
; RUN: opt -aa-pipeline=basic-aa -passes=attributor -attributor-manifest-internal -attributor-annotate-decl-cs -S <
%s | FileCheck %s --check-prefixes=CHECK,TUNIT
; RUN: opt -aa-pipeline=basic-aa -passes=attributor-cgscscc -attributor-manifest-internal -attributor-annotate-decl-cs -
S < %s | FileCheck %s --check-prefixes=CHECK,CGSCC
; Test that we only promote arguments when the caller/callee have compatible
; function attributes.

target triple = "x86_64-unknown-linux-gnu"

; This should promote
define internal fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512(ptr %arg, ptr
readonly %arg1) #0 {

```

```

;
; CHECK: Function Attrs: inlinehint mustprogress norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CHECK-LABEL: define {{^(^@)+}}@callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512
; CHECK-SAME:
(ptr noalias nocapture norecurse noundef nonnull writeonly align 64 dereferenceable(64) [[ARG:%.*]], <8 x i64>
[[TMP0:%.*]]) #[[ATTR0:[0-9]+]] {
; CHECK-NEXT: bb:
; CHECK-NEXT: [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
; CHECK-NEXT: store <8 x i64> [[TMP0]], ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT: [[TMP:%.*]] = load <8 x i64>, ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT: store <8 x i64> [[TMP]], ptr [[ARG]], align 64
; CHECK-NEXT: ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

define void @avx512_legal512_prefer512_call_avx512_legal512_prefer512(ptr %arg) #0 {
;
; TUNIT: Function Attrs: inlinehint mustprogress norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; TUNIT-LABEL: define {{^(^@)+}}@avx512_legal512_prefer512_call_avx512_legal512_prefer512
; TUNIT-SAME: (ptr nocapture norecurse writeonly [[ARG:%.*]]) #[[ATTR0]] {
; TUNIT-NEXT: bb:
; TUNIT-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align
32
; TUNIT-NEXT: [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT: call void @llvm.memset.p0.i64(ptr noalias nocapture norecurse noundef nonnull writeonly align 32
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5:[0-9]+]]
; TUNIT-NEXT: [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64
; TUNIT-NEXT: call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512(ptr
noalias nocapture norecurse noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64>
[[TMP0]]) #[[ATTR6:[0-9]+]]
; TUNIT-NEXT: [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; TUNIT-NEXT: store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; TUNIT-NEXT: ret void
;
; CGSCC: Function Attrs: inlinehint mustprogress norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CGSCC-LABEL: define {{^(^@)+}}@avx512_legal512_prefer512_call_avx512_legal512_prefer512
; CGSCC-SAME: (ptr nocapture
norecurse noundef nonnull writeonly align 2 dereferenceable(64) [[ARG:%.*]]) #[[ATTR0]] {
; CGSCC-NEXT: bb:
; CGSCC-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT: [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32

```

```

; CGSCC-NEXT:  call void @llvm.memset.p0.i64(ptr noalias nocapture nofree noundef nonnull writeonly align 64
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5:[0-9]+]]
; CGSCC-NEXT:  [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64
; CGSCC-NEXT:  call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512(ptr
noalias nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64>
[[TMP0]]) #[[ATTR6:[0-9]+]]
; CGSCC-NEXT:  [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; CGSCC-NEXT:  store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; CGSCC-NEXT:  ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align 32
call void
@llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512(ptr %tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

; This should promote
define internal fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256(ptr %arg, ptr
readonly %arg1) #1 {
;
; CHECK: Function Attrs: inlinehint mustprogress nofree norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CHECK-LABEL: define {{^[^@]+}}@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256
; CHECK-SAME: (ptr noalias nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64)
[[ARG:%.*]], <8 x i64> [[TMP0:%.*]]) #[[ATTR1:[0-9]+]] {
; CHECK-NEXT: bb:
; CHECK-NEXT:  [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
; CHECK-NEXT:  store <8 x i64> [[TMP0]], ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT:  [[TMP:%.*]] = load <8 x
i64>, ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT:  store <8 x i64> [[TMP]], ptr [[ARG]], align 64
; CHECK-NEXT:  ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

define void @avx512_legal512_prefer256_call_avx512_legal512_prefer256(ptr %arg) #1 {
;
; TUNIT: Function Attrs: inlinehint mustprogress nofree norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable

```

```

; TUNIT-LABEL: define {{^[^@]+}}@avx512_legal512_prefer256_call_avx512_legal512_prefer256
; TUNIT-SAME: (ptr nocapture norefree writeonly [[ARG:%.*]]) #[[ATTR1]] {
; TUNIT-NEXT: bb:
; TUNIT-NEXT:  [[TMP:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT:  [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT:  call void @llvm.memset.p0.i64(ptr noalias nocapture norefree noundef nonnull writeonly align 32
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5]]
; TUNIT-NEXT:  [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64
; TUNIT-NEXT:
    call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256(ptr noalias nocapture
norefree noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64> [[TMP0]]) #[[ATTR6]]
; TUNIT-NEXT:  [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; TUNIT-NEXT:  store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; TUNIT-NEXT:  ret void
;
; CGSCC: Function Attrs: inlinehint mustprogress norefree norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CGSCC-LABEL: define {{^[^@]+}}@avx512_legal512_prefer256_call_avx512_legal512_prefer256
; CGSCC-SAME: (ptr nocapture norefree noundef nonnull writeonly align 2 dereferenceable(64) [[ARG:%.*]])
#[[ATTR1]] {
; CGSCC-NEXT: bb:
; CGSCC-NEXT:  [[TMP:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT:  [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT:  call void @llvm.memset.p0.i64(ptr noalias nocapture norefree noundef nonnull writeonly align 64
dereferenceable(64)
[[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5]]
; CGSCC-NEXT:  [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64
; CGSCC-NEXT:  call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256(ptr
noalias nocapture norefree noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64>
[[TMP0]]) #[[ATTR6]]
; CGSCC-NEXT:  [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; CGSCC-NEXT:  store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; CGSCC-NEXT:  ret void
;
bb:
    %tmp = alloca <8 x i64>, align 32
    %tmp2 = alloca <8 x i64>, align 32
    call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
    call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256(ptr %tmp2, ptr %tmp)
    %tmp4 = load <8 x i64>, ptr %tmp2, align 32
    store <8 x i64> %tmp4, ptr %arg, align 2
    ret void
}

; This should promote
define internal fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256(ptr
%arg, ptr readonly %arg1) #1 {
;

```

```

; CHECK: Function Attrs: inlinehint mustprogress nofree norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CHECK-LABEL: define {{^(^@)+}}@callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256
; CHECK-SAME: (ptr noalias nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64)
[[ARG:%.*]], <8 x i64> [[TMP0:%.*]]) #[[ATTR1]] {
; CHECK-NEXT: bb:
; CHECK-NEXT: [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
; CHECK-NEXT: store <8 x i64> [[TMP0]], ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT: [[TMP:%.*]] = load <8 x i64>, ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT: store <8 x i64> [[TMP]], ptr [[ARG]], align 64
; CHECK-NEXT: ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

define void @avx512_legal512_prefer512_call_avx512_legal512_prefer256(ptr %arg) #0 {
;
; TUNIT: Function Attrs: inlinehint mustprogress nofree norecurse nosync nounwind
willreturn memory(argmem: readwrite) uwtable
; TUNIT-LABEL: define {{^(^@)+}}@avx512_legal512_prefer512_call_avx512_legal512_prefer256
; TUNIT-SAME: (ptr nocapture nofree writeonly [[ARG:%.*]]) #[[ATTR0]] {
; TUNIT-NEXT: bb:
; TUNIT-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT: [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT: call void @llvm.memset.p0.i64(ptr noalias nocapture nofree noundef nonnull writeonly align 32
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5]]
; TUNIT-NEXT: [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64
; TUNIT-NEXT: call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256(ptr
noalias nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64>
[[TMP0]]) #[[ATTR6]]
; TUNIT-NEXT: [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; TUNIT-NEXT: store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; TUNIT-NEXT:
ret void
;
; CGSCC: Function Attrs: inlinehint mustprogress nofree norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CGSCC-LABEL: define {{^(^@)+}}@avx512_legal512_prefer512_call_avx512_legal512_prefer256
; CGSCC-SAME: (ptr nocapture nofree noundef nonnull writeonly align 2 dereferenceable(64) [[ARG:%.*]])
#[[ATTR0]] {
; CGSCC-NEXT: bb:
; CGSCC-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT: [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT: call void @llvm.memset.p0.i64(ptr noalias nocapture nofree noundef nonnull writeonly align 64
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5]]

```

```

; CGSCC-NEXT:  [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64
; CGSCC-NEXT:  call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256(ptr
noalias nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64>
[[TMP0]]) #[[ATTR6]]
; CGSCC-NEXT:
    [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; CGSCC-NEXT:  store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; CGSCC-NEXT:  ret void
;
bb:
    %tmp = alloca <8 x i64>, align 32
    %tmp2 = alloca <8 x i64>, align 32
    call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
    call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256(ptr %tmp2, ptr %tmp)
    %tmp4 = load <8 x i64>, ptr %tmp2, align 32
    store <8 x i64> %tmp4, ptr %arg, align 2
    ret void
}

; This should promote
define internal fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512(ptr %arg, ptr
readonly %arg1) #0 {
;
; CHECK: Function Attrs: inlinehint mustprogress nofree norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CHECK-LABEL: define {{^[^@]+}}@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512
; CHECK-SAME: (ptr noalias nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64)
[[ARG:%.*]],
    <8 x i64> [[TMP0:%.*]]) #[[ATTR0]] {
; CHECK-NEXT: bb:
; CHECK-NEXT:  [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
; CHECK-NEXT:  store <8 x i64> [[TMP0]], ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT:  [[TMP:%.*]] = load <8 x i64>, ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT:  store <8 x i64> [[TMP]], ptr [[ARG]], align 64
; CHECK-NEXT:  ret void
;
bb:
    %tmp = load <8 x i64>, ptr %arg1
    store <8 x i64> %tmp, ptr %arg
    ret void
}

define void @avx512_legal512_prefer256_call_avx512_legal512_prefer512(ptr %arg) #1 {
;
; TUNIT: Function Attrs: inlinehint mustprogress nofree norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; TUNIT-LABEL: define {{^[^@]+}}@avx512_legal512_prefer256_call_avx512_legal512_prefer512
; TUNIT-SAME: (ptr nocapture nofree writeonly [[ARG:%.*]]) #[[ATTR1]] {

```

```

; TUNIT-NEXT: bb:
; TUNIT-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT: [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT: call void @llvm.memset.p0.i64(ptr
noalias nocapture nofree noundef nonnull writeonly align 32 dereferenceable(64) [[TMP]], i8 noundef 0, i64
noundef 32, i1 noundef false) #[[ATTR5]]
; TUNIT-NEXT: [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64
; TUNIT-NEXT: call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512(ptr
noalias nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64>
[[TMP0]]) #[[ATTR6]]
; TUNIT-NEXT: [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; TUNIT-NEXT: store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; TUNIT-NEXT: ret void
;
; CGSCC: Function Attrs: inlinehint mustprogress nofree norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CGSCC-LABEL: define { {^@}+ } @avx512_legal512_prefer256_call_avx512_legal512_prefer512
; CGSCC-SAME: (ptr nocapture nofree noundef nonnull writeonly align 2 dereferenceable(64) [[ARG:%.*]])
#[[ATTR1]] {
; CGSCC-NEXT: bb:
; CGSCC-NEXT:
[[TMP:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT: [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT: call void @llvm.memset.p0.i64(ptr noalias nocapture nofree noundef nonnull writeonly align 64
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5]]
; CGSCC-NEXT: [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64
; CGSCC-NEXT: call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512(ptr
noalias nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64>
[[TMP0]]) #[[ATTR6]]
; CGSCC-NEXT: [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; CGSCC-NEXT: store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; CGSCC-NEXT: ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align 32
call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512(ptr
%tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

; This should not promote
define internal fastcc void @callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr %arg, ptr
readonly %arg1) #1 {
;

```



```

; CHECK: Function Attrs: inlinehint mustprogress norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CHECK-LABEL: define {{^(^@)+}}@callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256
; CHECK-SAME: (ptr noalias nocapture norecurse noundef nonnull writeonly align 64 dereferenceable(64)
[[ARG:%.*]], ptr noalias nocapture norecurse noundef nonnull readonly align 64 dereferenceable(64) [[ARG1:%.*]])
#[[ATTR1]] {
; CHECK-NEXT: bb:
; CHECK-NEXT: [[TMP:%.*]] = load <8 x i64>, ptr [[ARG1]], align 64
; CHECK-NEXT: store <8 x i64> [[TMP]], ptr [[ARG]], align 64
; CHECK-NEXT: ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

define void @avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr %arg) #2 {
;
; TUNIT: Function Attrs: inlinehint mustprogress norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; TUNIT-LABEL: define {{^(^@)+}}@avx512_legal256_prefer256_call_avx512_legal512_prefer256
; TUNIT-SAME: (ptr nocapture norecurse writeonly [[ARG:%.*]]) #[[ATTR2:[0-9]+]] {
; TUNIT-NEXT: bb:
; TUNIT-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT: [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT: call void @llvm.memset.p0.i64(ptr noalias nocapture norecurse noundef nonnull writeonly align 32
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5]]
; TUNIT-NEXT: call fastcc void @callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr
noalias nocapture norecurse noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], ptr noalias
nocapture norecurse noundef nonnull readonly align 64 dereferenceable(64)
[[TMP]]) #[[ATTR6]]
; TUNIT-NEXT: [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; TUNIT-NEXT: store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; TUNIT-NEXT: ret void
;
; CGSCC: Function Attrs: inlinehint mustprogress norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CGSCC-LABEL: define {{^(^@)+}}@avx512_legal256_prefer256_call_avx512_legal512_prefer256
; CGSCC-SAME: (ptr nocapture norecurse noundef nonnull writeonly align 2 dereferenceable(64) [[ARG:%.*]])
#[[ATTR2:[0-9]+]] {
; CGSCC-NEXT: bb:
; CGSCC-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT: [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT: call void @llvm.memset.p0.i64(ptr noalias nocapture norecurse noundef nonnull writeonly align 64
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5]]
; CGSCC-NEXT: call fastcc void @callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr
noalias

```

```

    nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], ptr noalias nocapture
    nofree noundef nonnull readonly align 64 dereferenceable(64) [[TMP]]) #[[ATTR6]]
; CGSCC-NEXT:  [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; CGSCC-NEXT:  store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; CGSCC-NEXT:  ret void
;
bb:
    %tmp = alloca <8 x i64>, align 32
    %tmp2 = alloca <8 x i64>, align 32
    call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
    call fastcc void @callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr %tmp2, ptr %tmp)
    %tmp4 = load <8 x i64>, ptr %tmp2, align 32
    store <8 x i64> %tmp4, ptr %arg, align 2
    ret void
}

; This should not promote
define internal fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr %arg, ptr
readonly %arg1) #2 {
;
; CHECK: Function Attrs: inlinehint mustprogress nofree norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
;
    CHECK-LABEL: define {{[^@]+}}@callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256
; CHECK-SAME: (ptr noalias nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64)
[[ARG:%.*]], ptr noalias nocapture nofree noundef nonnull readonly align 64 dereferenceable(64) [[ARG1:%.*]])
#[[ATTR2:[0-9]+]] {
; CHECK-NEXT: bb:
; CHECK-NEXT:  [[TMP:%.*]] = load <8 x i64>, ptr [[ARG1]], align 64
; CHECK-NEXT:  store <8 x i64> [[TMP]], ptr [[ARG]], align 64
; CHECK-NEXT:  ret void
;
bb:
    %tmp = load <8 x i64>, ptr %arg1
    store <8 x i64> %tmp, ptr %arg
    ret void
}

define void @avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr %arg) #1 {
;
; TUNIT: Function Attrs: inlinehint mustprogress nofree norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; TUNIT-LABEL: define {{[^@]+}}@avx512_legal512_prefer256_call_avx512_legal256_prefer256
; TUNIT-SAME: (ptr nocapture nofree writeonly [[ARG:%.*]]) #[[ATTR1]] {
; TUNIT-NEXT: bb:
; TUNIT-NEXT:
    [[TMP:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT:  [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32

```

```

; TUNIT-NEXT: call void @llvm.memset.p0.i64(ptr noalias nocapture norecurse noalias noundef nonnull writeonly align 32
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5]]
; TUNIT-NEXT: call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr noalias
nocapture norecurse noalias noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], ptr noalias
nocapture norecurse noalias noundef nonnull readonly align 64 dereferenceable(64) [[TMP]]) #[[ATTR6]]
; TUNIT-NEXT: [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; TUNIT-NEXT: store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; TUNIT-NEXT: ret void
;
; CGSCC: Function Attrs: inlinehint mustprogress norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CGSCC-LABEL: define {{^@+}}@avx512_legal512_prefer256_call_avx512_legal256_prefer256
;
CGSCC-SAME: (ptr nocapture norecurse noalias noundef nonnull writeonly align 2 dereferenceable(64) [[ARG:%.*]])
#[[ATTR1]] {
; CGSCC-NEXT: bb:
; CGSCC-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT: [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT: call void @llvm.memset.p0.i64(ptr noalias nocapture norecurse noalias noundef nonnull writeonly align 64
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5]]
; CGSCC-NEXT: call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr noalias
nocapture norecurse noalias noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], ptr noalias
nocapture norecurse noalias noundef nonnull readonly align 64 dereferenceable(64) [[TMP]]) #[[ATTR6]]
; CGSCC-NEXT: [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; CGSCC-NEXT: store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; CGSCC-NEXT: ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>,
align 32
call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr %tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

; This should promote
define internal fastcc void @callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256(ptr %arg, ptr readonly
%arg1) #3 {
;
; CHECK: Function Attrs: inlinehint mustprogress norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CHECK-LABEL: define {{^@+}}@callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256
; CHECK-SAME: (ptr noalias nocapture norecurse noalias noundef nonnull writeonly align 64 dereferenceable(64)
[[ARG:%.*]], <8 x i64> [[TMP0:%.*]]) #[[ATTR3:[0-9]+]] {
; CHECK-NEXT: bb:

```

```

; CHECK-NEXT: [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
; CHECK-NEXT: store <8 x i64> [[TMP0]], ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT: [[TMP:%.*]]
= load <8 x i64>, ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT: store <8 x i64> [[TMP]], ptr [[ARG]], align 64
; CHECK-NEXT: ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

define void @avx2_legal256_prefer256_call_avx2_legal512_prefer256(ptr %arg) #4 {
;
; TUNIT: Function Attrs: inlinehint mustprogress norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; TUNIT-LABEL: define { { (^@)+ } } @avx2_legal256_prefer256_call_avx2_legal512_prefer256
; TUNIT-SAME: (ptr nocapture norecurse writeonly [[ARG:%.*]]) # [[ATTR3]] {
; TUNIT-NEXT: bb:
; TUNIT-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT: [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT: call void @llvm.memset.p0.i64(ptr noalias nocapture norecurse noundef nonnull writeonly align 32
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) # [[ATTR5]]
; TUNIT-NEXT: [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64
;
TUNIT-NEXT: call fastcc void @callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256(ptr noalias
nocapture norecurse noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64> [[TMP0]])
# [[ATTR6]]
; TUNIT-NEXT: [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; TUNIT-NEXT: store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; TUNIT-NEXT: ret void
;
; CGSCC: Function Attrs: inlinehint mustprogress norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CGSCC-LABEL: define { { (^@)+ } } @avx2_legal256_prefer256_call_avx2_legal512_prefer256
; CGSCC-SAME: (ptr nocapture norecurse noundef nonnull writeonly align 2 dereferenceable(64) [[ARG:%.*]])
# [[ATTR3]] {
; CGSCC-NEXT: bb:
; CGSCC-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT: [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT: call void @llvm.memset.p0.i64(ptr noalias nocapture norecurse noundef nonnull writeonly align 64
dereferenceable(64) [[TMP]],
i8 noundef 0, i64 noundef 32, i1 noundef false) # [[ATTR5]]
; CGSCC-NEXT: [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64
; CGSCC-NEXT: call fastcc void @callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256(ptr noalias
nocapture norecurse noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64> [[TMP0]])
# [[ATTR6]]

```

```

; CGSCC-NEXT:  [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; CGSCC-NEXT:  store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; CGSCC-NEXT:  ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align 32
call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
call fastcc void @callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256(ptr %tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

; This should promote
define internal fastcc void @callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256(ptr
%arg, ptr readonly %arg1) #4 {
;
; CHECK: Function Attrs: inlinehint mustprogress norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; CHECK-LABEL: define {{^[^@]+}}@callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256
; CHECK-SAME: (ptr noalias nocapture norecurse noundef nonnull writeonly align 64 dereferenceable(64)
[[ARG:%.*]], <8 x i64> [[TMP0:%.*]]) #[[ATTR3]] {
; CHECK-NEXT: bb:
; CHECK-NEXT:  [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
; CHECK-NEXT:  store <8 x i64> [[TMP0]], ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT:  [[TMP:%.*]] = load <8 x i64>, ptr [[ARG1_PRIV]], align 64
; CHECK-NEXT:  store <8 x i64> [[TMP]], ptr [[ARG]], align 64
; CHECK-NEXT:  ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

define void @avx2_legal512_prefer256_call_avx2_legal256_prefer256(ptr %arg) #3 {
;
; TUNIT: Function Attrs: inlinehint mustprogress norecurse nosync nounwind willreturn memory(argmem:
readwrite) uwtable
; TUNIT-LABEL: define {{^[^@]+}}@avx2_legal512_prefer256_call_avx2_legal256_prefer256
; TUNIT-SAME: (ptr nocapture norecurse writeonly [[ARG:%.*]]) #[[ATTR3]] {
; TUNIT-NEXT: bb:
; TUNIT-NEXT:  [[TMP:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT:  [[TRUETMP2:%.*]] = alloca <8 x i64>, align 32
; TUNIT-NEXT:  call void @llvm.memset.p0.i64(ptr noalias nocapture norecurse noundef nonnull writeonly align 32
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5]]
; TUNIT-NEXT:  [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64

```

```

; TUNIT-NEXT:  call fastcc void @callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256(ptr noalias
nocapture norecurse noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64> [[TMP0]])
#[[ATTR6]]
; TUNIT-NEXT:  [[TRUETMP4:%.*]] = load <8 x i64>, ptr [[TRUETMP2]], align 64
; TUNIT-NEXT:  store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; TUNIT-NEXT:  ret void
;
; CGSCC: Function
Attrs: inlinehint mustprogress norecurse noundef nonnull writeonly align 2 dereferenceable(64) [[ARG:%.*]]
; CGSCC-LABEL: define { {^@+} } @avx2_legal512_prefer256_call_avx2_legal256_prefer256
; CGSCC-SAME: (ptr nocapture norecurse noundef nonnull writeonly align 2 dereferenceable(64) [[ARG:%.*]])
#[[ATTR3]] {
; CGSCC-NEXT: bb:
; CGSCC-NEXT:  [[TMP:%.*]] = alloca <8 x i64>, align 32
; CGSCC-NEXT:  [[TRUETMP2:%.*]] = alloca <8 x i64>, align 64
; CGSCC-NEXT:  call void @llvm.memset.p0.i64(ptr noalias nocapture norecurse noundef nonnull writeonly align 64
dereferenceable(64) [[TMP]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR5]]
; CGSCC-NEXT:  [[TMP0:%.*]] = load <8 x i64>, ptr [[TMP]], align 64
; CGSCC-NEXT:  call fastcc void @callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256(ptr noalias
nocapture norecurse noundef nonnull writeonly align 64 dereferenceable(64) [[TRUETMP2]], <8 x i64> [[TMP0]])
#[[ATTR6]]
; CGSCC-NEXT:  [[TRUETMP4:%.*]] = load <8 x i64>,
ptr [[TRUETMP2]], align 64
; CGSCC-NEXT:  store <8 x i64> [[TRUETMP4]], ptr [[ARG]], align 2
; CGSCC-NEXT:  ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align 32
call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
call fastcc void @callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256(ptr %tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

```

```

; Function Attrs: argmemonly nounwind
declare void @llvm.memset.p0.i64(ptr nocapture writeonly, i8, i64, i1) #5

```

```

attributes #0 = { inlinehint norecurse nounwind uwtable "target-features"="+avx512vl" "min-legal-vector-
width"="512" "prefer-vector-width"="512" }
attributes #1 = { inlinehint norecurse nounwind uwtable "target-features"="+avx512vl" "min-legal-vector-
width"="512" "prefer-vector-width"="256" }
attributes #2 = { inlinehint norecurse nounwind uwtable "target-features"="+avx512vl" "min-legal-vector-
width"="256"
"prefer-vector-width"="256" }
attributes #3 = { inlinehint norecurse nounwind uwtable "target-features"="+avx2" "min-legal-vector-width"="512"
"prefer-vector-width"="256" }

```

```

attributes #4 = { inlinehint norecurse nounwind uwtable "target-features"="+avx2" "min-legal-vector-width"="256"
"prefer-vector-width"="256" }
attributes #5 = { argmemonly nounwind }
;
; TUNIT: attributes #[[ATTR0]] = { inlinehint mustprogress nofree norecurse nosync nounwind willreturn
memory(argmem: readwrite) uwtable "min-legal-vector-width"="512" "prefer-vector-width"="512" "target-
features"="+avx512vl" }
; TUNIT: attributes #[[ATTR1]] = { inlinehint mustprogress nofree norecurse nosync nounwind willreturn
memory(argmem: readwrite) uwtable "min-legal-vector-width"="512" "prefer-vector-width"="256" "target-
features"="+avx512vl" }
; TUNIT: attributes #[[ATTR2]] = { inlinehint mustprogress nofree norecurse nosync nounwind willreturn
memory(argmem: readwrite) uwtable "min-legal-vector-width"="256" "prefer-vector-width"="256"
"target-features"="+avx512vl" }
; TUNIT: attributes #[[ATTR3]] = { inlinehint mustprogress nofree norecurse nosync nounwind willreturn
memory(argmem: readwrite) uwtable "min-legal-vector-width"="512" "prefer-vector-width"="256" "target-
features"="+avx2" }
; TUNIT: attributes #[[ATTR4:[0-9]+]] = { nocallback nofree nounwind willreturn memory(argmem: write) }
; TUNIT: attributes #[[ATTR5]] = { nofree willreturn memory(write) }
; TUNIT: attributes #[[ATTR6]] = { nofree nosync nounwind willreturn }
;
; CGSCC: attributes #[[ATTR0]] = { inlinehint mustprogress nofree norecurse nosync nounwind willreturn
memory(argmem: readwrite) uwtable "min-legal-vector-width"="512" "prefer-vector-width"="512" "target-
features"="+avx512vl" }
; CGSCC: attributes #[[ATTR1]] = { inlinehint mustprogress nofree norecurse nosync nounwind willreturn
memory(argmem: readwrite) uwtable "min-legal-vector-width"="512" "prefer-vector-width"="256" "target-
features"="+avx512vl" }
; CGSCC: attributes #[[ATTR2]]
= { inlinehint mustprogress nofree norecurse nosync nounwind willreturn memory(argmem: readwrite) uwtable
"min-legal-vector-width"="256" "prefer-vector-width"="256" "target-features"="+avx512vl" }
; CGSCC: attributes #[[ATTR3]] = { inlinehint mustprogress nofree norecurse nosync nounwind willreturn
memory(argmem: readwrite) uwtable "min-legal-vector-width"="512" "prefer-vector-width"="256" "target-
features"="+avx2" }
; CGSCC: attributes #[[ATTR4:[0-9]+]] = { nocallback nofree nounwind willreturn memory(argmem: write) }
; CGSCC: attributes #[[ATTR5]] = { nofree willreturn memory(write) }
; CGSCC: attributes #[[ATTR6]] = { nofree nounwind willreturn }
;
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64I
; RUN: llc -mtriple=riscv64 -mattr=+zbb -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64ZBB

declare i32 @llvm.ctlz.i32(i32, i1)

define signext i32 @ctlz_i32(i32 signext %a) nounwind {
; RV64I-LABEL: ctlz_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT: beqz a0, .LBB0_2

```

```

; RV64I-NEXT: # %bb.1: # %cond.false
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  srliw a1, a0, 1
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 2
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 4
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 8
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 16
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:
    not a0, a0
; RV64I-NEXT:  srliw a1, a0, 1
; RV64I-NEXT:  lui a2, 349525
; RV64I-NEXT:  addi a2, a2, 1365
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  subw a0, a0, a1
; RV64I-NEXT:  lui a1, 209715
; RV64I-NEXT:  addi a1, a1, 819
; RV64I-NEXT:  and a2, a0, a1
; RV64I-NEXT:  srliw a0, a0, 2
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  add a0, a2, a0
; RV64I-NEXT:  srliw a1, a0, 4
; RV64I-NEXT:  addw a0, a0, a1
; RV64I-NEXT:  lui a1, 61681
; RV64I-NEXT:  addiw a1, a1, -241
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  lui a1, 4112
; RV64I-NEXT:  addiw a1, a1, 257
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:  srliw a0, a0, 24
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
; RV64I-NEXT: .LBB0_2:
; RV64I-NEXT:  li a0, 32
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: ctlz_i32:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT:  clzw a0, a0
; RV64ZBB-NEXT:  ret
%1 = call i32 @llvm.ctlz.i32(i32 %a, i1 false)

```



```

ret i32 %1
}

```

```

define signext i32 @log2_i32(i32 signext %a) nounwind {
; RV64I-LABEL: log2_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT: beqz a0, .LBB1_2
; RV64I-NEXT: # %bb.1: # %cond.false
; RV64I-NEXT: addi sp, sp, -16
; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT: srliw a1, a0, 1
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 2
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 4
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 8
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 16
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: not a0, a0
; RV64I-NEXT: srliw a1, a0, 1
; RV64I-NEXT: lui a2, 349525
; RV64I-NEXT: addi a2, a2, 1365
; RV64I-NEXT: and a1, a1, a2
; RV64I-NEXT: subw a0, a0, a1
; RV64I-NEXT: lui a1, 209715
; RV64I-NEXT: addi a1, a1, 819
; RV64I-NEXT: and a2, a0, a1
; RV64I-NEXT: srliw a0, a0, 2
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: add a0, a2, a0
; RV64I-NEXT:
; RV64I-NEXT: srliw a1, a0, 4
; RV64I-NEXT: addw a0, a0, a1
; RV64I-NEXT: lui a1, 61681
; RV64I-NEXT: addiw a1, a1, -241
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: lui a1, 4112
; RV64I-NEXT: addiw a1, a1, 257
; RV64I-NEXT: call __muldi3
; RV64I-NEXT: srliw a0, a0, 24
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT: addi sp, sp, 16
; RV64I-NEXT: j .LBB1_3
; RV64I-NEXT: .LBB1_2:
; RV64I-NEXT: li a0, 32
; RV64I-NEXT: .LBB1_3: # %cond.end

```

```

; RV64I-NEXT:  li a1, 31
; RV64I-NEXT:  subw a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: log2_i32:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  clzw a0, a0
; RV64ZBB-NEXT:  li a1, 31
; RV64ZBB-NEXT:  subw a0, a1, a0
; RV64ZBB-NEXT:  ret
%1 = call i32 @llvmctlz.i32(i32 %a, i1 false)
%2 = sub i32 31, %1
ret i32 %2
}

```

```

define signext i32 @log2_ceil_i32(i32 signext %a) nounwind {
; RV64I-LABEL: log2_ceil_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra,
; RV64I-NEXT:  8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  sd s0, 0(sp) # 8-byte Folded Spill
; RV64I-NEXT:  addiw a0, a0, -1
; RV64I-NEXT:  li s0, 32
; RV64I-NEXT:  li a1, 32
; RV64I-NEXT:  beqz a0, .LBB2_2
; RV64I-NEXT:  # %bb.1: # %cond.false
; RV64I-NEXT:  srliw a1, a0, 1
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 2
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 4
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 8
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 16
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  srliw a1, a0, 1
; RV64I-NEXT:  lui a2, 349525
; RV64I-NEXT:  addi a2, a2, 1365
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  subw a0, a0, a1
; RV64I-NEXT:  lui a1, 209715
; RV64I-NEXT:  addi a1, a1, 819
; RV64I-NEXT:  and a2, a0, a1
; RV64I-NEXT:  srliw a0, a0, 2
; RV64I-NEXT:  and a0, a0, a1

```

```

; RV64I-NEXT:  add a0, a2, a0
; RV64I-NEXT:  srliw a1, a0, 4
; RV64I-NEXT:
    addw a0, a0, a1
; RV64I-NEXT:  lui a1, 61681
; RV64I-NEXT:  addiw a1, a1, -241
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  lui a1, 4112
; RV64I-NEXT:  addiw a1, a1, 257
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:  srliw a1, a0, 24
; RV64I-NEXT:  .LBB2_2: # %cond.end
; RV64I-NEXT:  subw a0, s0, a1
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  ld s0, 0(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: log2_ceil_i32:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  addi a0, a0, -1
; RV64ZBB-NEXT:  clzw a0, a0
; RV64ZBB-NEXT:  li a1, 32
; RV64ZBB-NEXT:  subw a0, a1, a0
; RV64ZBB-NEXT:  ret
%1 = sub i32 %a, 1
%2 = call i32 @llvmctlz.i32(i32 %1, i1 false)
%3 = sub i32 32, %2
ret i32 %3
}

define signext i32 @findLastSet_i32(i32 signext %a) nounwind {
; RV64I-LABEL: findLastSet_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
;
    RV64I-NEXT:  sd s0, 0(sp) # 8-byte Folded Spill
; RV64I-NEXT:  mv s0, a0
; RV64I-NEXT:  srliw a0, a0, 1
; RV64I-NEXT:  or a0, s0, a0
; RV64I-NEXT:  srliw a1, a0, 2
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 4
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 8
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 16

```

```

; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  srlw a1, a0, 1
; RV64I-NEXT:  lui a2, 349525
; RV64I-NEXT:  addi a2, a2, 1365
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  subw a0, a0, a1
; RV64I-NEXT:  lui a1, 209715
; RV64I-NEXT:  addi a1, a1, 819
; RV64I-NEXT:  and a2, a0, a1
; RV64I-NEXT:  srlw a0, a0, 2
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  add a0, a2, a0
; RV64I-NEXT:  srlw a1, a0, 4
; RV64I-NEXT:  addw a0, a0, a1
; RV64I-NEXT:  lui a1, 61681
; RV64I-NEXT:  addiw a1, a1, -241
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  lui a1, 4112
; RV64I-NEXT:
    addiw a1, a1, 257
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:  srlw a0, a0, 24
; RV64I-NEXT:  xori a0, a0, 31
; RV64I-NEXT:  snez a1, s0
; RV64I-NEXT:  addiw a1, a1, -1
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  ld s0, 0(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: findLastSet_i32:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  clzw a1, a0
; RV64ZBB-NEXT:  xori a1, a1, 31
; RV64ZBB-NEXT:  snez a0, a0
; RV64ZBB-NEXT:  addiw a0, a0, -1
; RV64ZBB-NEXT:  or a0, a0, a1
; RV64ZBB-NEXT:  ret
%1 = call i32 @llvmctlz.i32(i32 %a, i1 true)
%2 = xor i32 31, %1
%3 = icmp eq i32 %a, 0
%4 = select i1 %3, i32 -1, i32 %2
ret i32 %4
}

define i32 @ctlz_lshr_i32(i32 signext %a) {

```

```

; RV64I-LABEL: ctiz_lshr_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srlw a0, a0, 1
; RV64I-NEXT:  beqz a0, .LBB4_2
; RV64I-NEXT:  # %bb.1: # %cond.false
; RV64I-NEXT:  addi sp, sp,
-16
; RV64I-NEXT:  .cfi_def_cfa_offset 16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  .cfi_offset ra, -8
; RV64I-NEXT:  srlw a1, a0, 1
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srlw a1, a0, 2
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srlw a1, a0, 4
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srlw a1, a0, 8
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srlw a1, a0, 16
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  srlw a1, a0, 1
; RV64I-NEXT:  lui a2, 349525
; RV64I-NEXT:  addi a2, a2, 1365
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  subw a0, a0, a1
; RV64I-NEXT:  lui a1, 209715
; RV64I-NEXT:  addi a1, a1, 819
; RV64I-NEXT:  and a2, a0, a1
; RV64I-NEXT:  srlw a0, a0, 2
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  add a0, a2, a0
; RV64I-NEXT:  srlw a1, a0, 4
; RV64I-NEXT:  addw a0, a0, a1
; RV64I-NEXT:  lui a1, 61681
; RV64I-NEXT:  addiw a1, a1, -241
; RV64I-NEXT:  and a0,
a0, a1
; RV64I-NEXT:  lui a1, 4112
; RV64I-NEXT:  addiw a1, a1, 257
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:  srlw a0, a0, 24
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
; RV64I-NEXT: .LBB4_2:
; RV64I-NEXT:  li a0, 32
; RV64I-NEXT:  ret

```

```

;
; RV64ZBB-LABEL: ctlz_lshr_i32:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT: srliw a0, a0, 1
; RV64ZBB-NEXT: clzw a0, a0
; RV64ZBB-NEXT: ret
%1 = lshr i32 %a, 1
%2 = call i32 @llvm.ctlz.i32(i32 %1, i1 false)
ret i32 %2
}

declare i64 @llvm.ctlz.i64(i64, i1)

define i64 @ctlz_i64(i64 %a) nounwind {
; RV64I-LABEL: ctlz_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT: beqz a0, .LBB5_2
; RV64I-NEXT: # %bb.1: # %cond.false
; RV64I-NEXT: addi sp, sp, -16
; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT: srli a1, a0, 1
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srli a1, a0, 2
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srli a1, a0, 4
; RV64I-NEXT:
    or a0, a0, a1
; RV64I-NEXT: srli a1, a0, 8
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srli a1, a0, 16
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srli a1, a0, 32
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: not a0, a0
; RV64I-NEXT: srli a1, a0, 1
; RV64I-NEXT: lui a2, 349525
; RV64I-NEXT: addiw a2, a2, 1365
; RV64I-NEXT: slli a3, a2, 32
; RV64I-NEXT: add a2, a2, a3
; RV64I-NEXT: and a1, a1, a2
; RV64I-NEXT: sub a0, a0, a1
; RV64I-NEXT: lui a1, 209715
; RV64I-NEXT: addiw a1, a1, 819
; RV64I-NEXT: slli a2, a1, 32
; RV64I-NEXT: add a1, a1, a2
; RV64I-NEXT: and a2, a0, a1
; RV64I-NEXT: srli a0, a0, 2
; RV64I-NEXT: and a0, a0, a1

```

```

; RV64I-NEXT: add a0, a2, a0
; RV64I-NEXT: srli a1, a0, 4
; RV64I-NEXT: add a0, a0, a1
; RV64I-NEXT: lui a1, 61681
; RV64I-NEXT: addiw a1, a1, -241
; RV64I-NEXT: slli a2, a1, 32
; RV64I-NEXT: add a1, a1, a2
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: lui a1, 4112
; RV64I-NEXT:
    addiw a1, a1, 257
; RV64I-NEXT: slli a2, a1, 32
; RV64I-NEXT: add a1, a1, a2
; RV64I-NEXT: call __muldi3
; RV64I-NEXT: srli a0, a0, 56
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT: addi sp, sp, 16
; RV64I-NEXT: ret
; RV64I-NEXT: .LBB5_2:
; RV64I-NEXT: li a0, 64
; RV64I-NEXT: ret
;
; RV64ZBB-LABEL: ctlz_i64:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT: clz a0, a0
; RV64ZBB-NEXT: ret
%1 = call i64 @llvm.ctlz.i64(i64 %a, i1 false)
ret i64 %1
}

```

```

declare i32 @llvm.cttz.i32(i32, i1)

```

```

define signext i32 @cttz_i32(i32 signext %a) nounwind {
; RV64I-LABEL: cttz_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT: beqz a0, .LBB6_2
; RV64I-NEXT: # %bb.1: # %cond.false
; RV64I-NEXT: addi sp, sp, -16
; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT: negw a1, a0
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: lui a1, 30667
; RV64I-NEXT: addiw a1, a1, 1329
; RV64I-NEXT: call __muldi3
; RV64I-NEXT: srliw a0, a0, 27
;
    RV64I-NEXT: lui a1, %hi(.LCPI6_0)
; RV64I-NEXT: addi a1, a1, %lo(.LCPI6_0)

```

```

; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  lbu a0, 0(a0)
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
; RV64I-NEXT:  .LBB6_2:
; RV64I-NEXT:  li a0, 32
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: cttz_i32:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  ctzw a0, a0
; RV64ZBB-NEXT:  ret
%1 = call i32 @llvm.cttz.i32(i32 %a, i1 false)
ret i32 %1
}

define signext i32 @cttz_zero_undef_i32(i32 signext %a) nounwind {
; RV64I-LABEL: cttz_zero_undef_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  negw a1, a0
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  lui a1, 30667
; RV64I-NEXT:  addiw a1, a1, 1329
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:  srlw a0, a0, 27
; RV64I-NEXT:  lui a1, %hi(.LCPI7_0)
; RV64I-NEXT:  addi a1, a1, %lo(.LCPI7_0)
; RV64I-NEXT:
    add a0, a1, a0
; RV64I-NEXT:  lbu a0, 0(a0)
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: cttz_zero_undef_i32:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  ctzw a0, a0
; RV64ZBB-NEXT:  ret
%1 = call i32 @llvm.cttz.i32(i32 %a, i1 true)
ret i32 %1
}

define signext i32 @findFirstSet_i32(i32 signext %a) nounwind {
; RV64I-LABEL: findFirstSet_i32:
; RV64I:      # %bb.0:

```



```

; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  sd s0, 0(sp) # 8-byte Folded Spill
; RV64I-NEXT:  mv s0, a0
; RV64I-NEXT:  negw a0, a0
; RV64I-NEXT:  and a0, s0, a0
; RV64I-NEXT:  lui a1, 30667
; RV64I-NEXT:  addiw a1, a1, 1329
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:  srliw a0, a0, 27
; RV64I-NEXT:  lui a1, %hi(.LCPI8_0)
; RV64I-NEXT:  addi a1, a1, %lo(.LCPI8_0)
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  lbu a0, 0(a0)
; RV64I-NEXT:  snez a1, s0
; RV64I-NEXT:
    addi a1, a1, -1
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  ld s0, 0(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: findFirstSet_i32:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  ctzw a1, a0
; RV64ZBB-NEXT:  snez a0, a0
; RV64ZBB-NEXT:  addiw a0, a0, -1
; RV64ZBB-NEXT:  or a0, a0, a1
; RV64ZBB-NEXT:  ret
%1 = call i32 @llvm.cttz.i32(i32 %a, i1 true)
%2 = icmp eq i32 %a, 0
%3 = select i1 %2, i32 -1, i32 %1
ret i32 %3
}

define signext i32 @ffs_i32(i32 signext %a) nounwind {
; RV64I-LABEL: ffs_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  sd s0, 0(sp) # 8-byte Folded Spill
; RV64I-NEXT:  mv s0, a0
; RV64I-NEXT:  negw a0, a0
; RV64I-NEXT:  and a0, s0, a0
; RV64I-NEXT:  lui a1, 30667
; RV64I-NEXT:  addiw a1, a1, 1329
; RV64I-NEXT:  call __muldi3

```

```

; RV64I-NEXT:
    srlw a0, a0, 27
; RV64I-NEXT:  lui a1, %hi(.LCPI9_0)
; RV64I-NEXT:  addi a1, a1, %lo(.LCPI9_0)
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  lbu a0, 0(a0)
; RV64I-NEXT:  addi a0, a0, 1
; RV64I-NEXT:  seqz a1, s0
; RV64I-NEXT:  addi a1, a1, -1
; RV64I-NEXT:  and a0, a1, a0
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  ld s0, 0(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: ffs_i32:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  ctzw a1, a0
; RV64ZBB-NEXT:  addi a1, a1, 1
; RV64ZBB-NEXT:  seqz a0, a0
; RV64ZBB-NEXT:  addi a0, a0, -1
; RV64ZBB-NEXT:  and a0, a0, a1
; RV64ZBB-NEXT:  zext.h a0, a0
; RV64ZBB-NEXT:  ret
%1 = call i32 @llvm.cttz.i32(i32 %a, i1 true)
%2 = add i32 %1, 1
%3 = icmp eq i32 %a, 0
%4 = select i1 %3, i32 0, i32 %2
ret i32 %4
}

declare i64 @llvm.cttz.i64(i64, i1)

define i64 @cttz_i64(i64
    %a) nounwind {
; RV64I-LABEL: cttz_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  beqz a0, .LBB10_2
; RV64I-NEXT:  # %bb.1: # %cond.false
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  neg a1, a0
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  lui a1, %hi(.LCPI10_0)
; RV64I-NEXT:  ld a1, %lo(.LCPI10_0)(a1)
; RV64I-NEXT:  call __muldi3

```

```

; RV64I-NEXT: srli a0, a0, 58
; RV64I-NEXT: lui a1, %hi(.LCPI10_1)
; RV64I-NEXT: addi a1, a1, %lo(.LCPI10_1)
; RV64I-NEXT: add a0, a1, a0
; RV64I-NEXT: lbu a0, 0(a0)
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT: addi sp, sp, 16
; RV64I-NEXT: ret
; RV64I-NEXT: .LBB10_2:
; RV64I-NEXT: li a0, 64
; RV64I-NEXT: ret
;
; RV64ZBB-LABEL: cttz_i64:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT: ctz a0, a0
; RV64ZBB-NEXT: ret
%1 = call i64 @llvm.cttz.i64(i64 %a, i1 false)
ret i64 %1
}

```

```
declare i32 @llvm.ctpop.i32(i32)
```

```

define signext i32 @ctpop_i32(i32
  signext %a) nounwind {
; RV64I-LABEL: ctpop_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT: addi sp, sp, -16
; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT: srliw a1, a0, 1
; RV64I-NEXT: lui a2, 349525
; RV64I-NEXT: addi a2, a2, 1365
; RV64I-NEXT: and a1, a1, a2
; RV64I-NEXT: subw a0, a0, a1
; RV64I-NEXT: lui a1, 209715
; RV64I-NEXT: addi a1, a1, 819
; RV64I-NEXT: and a2, a0, a1
; RV64I-NEXT: srliw a0, a0, 2
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: add a0, a2, a0
; RV64I-NEXT: srliw a1, a0, 4
; RV64I-NEXT: addw a0, a0, a1
; RV64I-NEXT: lui a1, 61681
; RV64I-NEXT: addiw a1, a1, -241
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: lui a1, 4112
; RV64I-NEXT: addiw a1, a1, 257
; RV64I-NEXT: call __muldi3
; RV64I-NEXT: srliw a0, a0, 24

```

```

; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: ctpop_i32:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  cpopw
a0, a0
; RV64ZBB-NEXT:  ret
%1 = call i32 @llvm.ctpop.i32(i32 %a)
ret i32 %1
}

```

```

define signext i32 @ctpop_i32_load(ptr %p) nounwind {

```

```

; RV64I-LABEL: ctpop_i32_load:
; RV64I:        # %bb.0:
; RV64I-NEXT:   addi sp, sp, -16
; RV64I-NEXT:   sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:   lw a0, 0(a0)
; RV64I-NEXT:   srliw a1, a0, 1
; RV64I-NEXT:   lui a2, 349525
; RV64I-NEXT:   addi a2, a2, 1365
; RV64I-NEXT:   and a1, a1, a2
; RV64I-NEXT:   subw a0, a0, a1
; RV64I-NEXT:   lui a1, 209715
; RV64I-NEXT:   addi a1, a1, 819
; RV64I-NEXT:   and a2, a0, a1
; RV64I-NEXT:   srliw a0, a0, 2
; RV64I-NEXT:   and a0, a0, a1
; RV64I-NEXT:   add a0, a2, a0
; RV64I-NEXT:   srliw a1, a0, 4
; RV64I-NEXT:   addw a0, a0, a1
; RV64I-NEXT:   lui a1, 61681
; RV64I-NEXT:   addiw a1, a1, -241
; RV64I-NEXT:   and a0, a0, a1
; RV64I-NEXT:   lui a1, 4112
; RV64I-NEXT:   addiw a1, a1, 257
; RV64I-NEXT:   call __muldi3
; RV64I-NEXT:   srliw a0, a0, 24
; RV64I-NEXT:   ld ra, 8(sp)
# 8-byte Folded Reload
; RV64I-NEXT:   addi sp, sp, 16
; RV64I-NEXT:   ret
;
; RV64ZBB-LABEL: ctpop_i32_load:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  lw a0, 0(a0)
; RV64ZBB-NEXT:  cpopw a0, a0

```

```
; RV64ZBB-NEXT:  ret
%a = load i32, ptr %p
%1 = call i32 @llvm.ctpop.i32(i32 %a)
ret i32 %1
}
```

```
declare i64 @llvm.ctpop.i64(i64)
```

```
define i64 @ctpop_i64(i64 %a) nounwind {
; RV64I-LABEL: ctpop_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  srli a1, a0, 1
; RV64I-NEXT:  lui a2, 349525
; RV64I-NEXT:  addiw a2, a2, 1365
; RV64I-NEXT:  slli a3, a2, 32
; RV64I-NEXT:  add a2, a2, a3
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  sub a0, a0, a1
; RV64I-NEXT:  lui a1, 209715
; RV64I-NEXT:  addiw a1, a1, 819
; RV64I-NEXT:  slli a2, a1, 32
; RV64I-NEXT:  add a1, a1, a2
; RV64I-NEXT:  and a2, a0, a1
; RV64I-NEXT:  srli a0, a0, 2
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  add a0, a2,
a0
; RV64I-NEXT:  srli a1, a0, 4
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  lui a1, 61681
; RV64I-NEXT:  addiw a1, a1, -241
; RV64I-NEXT:  slli a2, a1, 32
; RV64I-NEXT:  add a1, a1, a2
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  lui a1, 4112
; RV64I-NEXT:  addiw a1, a1, 257
; RV64I-NEXT:  slli a2, a1, 32
; RV64I-NEXT:  add a1, a1, a2
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:  srli a0, a0, 56
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: ctpop_i64:
; RV64ZBB:    # %bb.0:
```

```

; RV64ZBB-NEXT:  cpop a0, a0
; RV64ZBB-NEXT:  ret
%1 = call i64 @llvm.ctpop.i64(i64 %a)
ret i64 %1
}

define signext i32 @sextb_i32(i32 signext %a) nounwind {
; RV64I-LABEL: sextb_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 56
; RV64I-NEXT:  srai a0, a0, 56
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: sextb_i32:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT:  sext.b a0, a0
; RV64ZBB-NEXT:
ret
%shl = shl i32 %a, 24
%shr = ashr exact i32 %shl, 24
ret i32 %shr
}

define i64 @sextb_i64(i64 %a) nounwind {
; RV64I-LABEL: sextb_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 56
; RV64I-NEXT:  srai a0, a0, 56
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: sextb_i64:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT:  sext.b a0, a0
; RV64ZBB-NEXT:  ret
%shl = shl i64 %a, 56
%shr = ashr exact i64 %shl, 56
ret i64 %shr
}

define signext i32 @sextb_i32(i32 signext %a) nounwind {
; RV64I-LABEL: sextb_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srai a0, a0, 48
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: sextb_i32:
; RV64ZBB:    # %bb.0:

```

```

; RV64ZBB-NEXT:  sext.h a0, a0
; RV64ZBB-NEXT:  ret
%shl = shl i32 %a, 16
%shr = ashr exact i32 %shl, 16
ret i32 %shr
}

define i64 @sexth_i64(i64 %a) nounwind {
; RV64I-LABEL: sexth_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srai a0, a0, 48
; RV64I-NEXT:
    ret
;
; RV64ZBB-LABEL: sexth_i64:
; RV64ZBB:     # %bb.0:
; RV64ZBB-NEXT:  sext.h a0, a0
; RV64ZBB-NEXT:  ret
%shl = shl i64 %a, 48
%shr = ashr exact i64 %shl, 48
ret i64 %shr
}

define signext i32 @min_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: min_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  blt a0, a1, .LBB18_2
; RV64I-NEXT:  # %bb.1:
; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT:  .LBB18_2:
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: min_i32:
; RV64ZBB:     # %bb.0:
; RV64ZBB-NEXT:  min a0, a0, a1
; RV64ZBB-NEXT:  ret
%cmp = icmp slt i32 %a, %b
%cond = select i1 %cmp, i32 %a, i32 %b
ret i32 %cond
}

define i64 @min_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: min_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  blt a0, a1, .LBB19_2
; RV64I-NEXT:  # %bb.1:
; RV64I-NEXT:  mv a0, a1

```

```

; RV64I-NEXT: .LBB19_2:
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: min_i64:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  min a0, a0, a1
; RV64ZBB-NEXT:  ret
%cmp = icmp
slt i64 %a, %b
%cond = select i1 %cmp, i64 %a, i64 %b
ret i64 %cond
}

define signext i32 @max_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: max_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  blt a1, a0, .LBB20_2
; RV64I-NEXT:  # %bb.1:
; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT:  .LBB20_2:
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: max_i32:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  max a0, a0, a1
; RV64ZBB-NEXT:  ret
%cmp = icmp sgt i32 %a, %b
%cond = select i1 %cmp, i32 %a, i32 %b
ret i32 %cond
}

define i64 @max_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: max_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  blt a1, a0, .LBB21_2
; RV64I-NEXT:  # %bb.1:
; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT:  .LBB21_2:
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: max_i64:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  max a0, a0, a1
; RV64ZBB-NEXT:  ret
%cmp = icmp sgt i64 %a, %b
%cond = select i1 %cmp, i64 %a, i64 %b
ret i64 %cond
}

```



```

define signext i32 @minu_i32(i32 signext
    %a, i32 signext %b) nounwind {
; RV64I-LABEL: minu_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  bltu a0, a1, .LBB22_2
; RV64I-NEXT: # %bb.1:
; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT: .LBB22_2:
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: minu_i32:
; RV64ZBB:     # %bb.0:
; RV64ZBB-NEXT: minu a0, a0, a1
; RV64ZBB-NEXT: ret
    %cmp = icmp ult i32 %a, %b
    %cond = select i1 %cmp, i32 %a, i32 %b
    ret i32 %cond
}

define i64 @minu_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: minu_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  bltu a0, a1, .LBB23_2
; RV64I-NEXT: # %bb.1:
; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT: .LBB23_2:
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: minu_i64:
; RV64ZBB:     # %bb.0:
; RV64ZBB-NEXT: minu a0, a0, a1
; RV64ZBB-NEXT: ret
    %cmp = icmp ult i64 %a, %b
    %cond = select i1 %cmp, i64 %a, i64 %b
    ret i64 %cond
}

define signext i32 @maxu_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: maxu_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  bltu a1,
    a0, .LBB24_2
; RV64I-NEXT: # %bb.1:
; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT: .LBB24_2:
; RV64I-NEXT:  ret
;

```

```

; RV64ZBB-LABEL: maxu_i32:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT:  maxu a0, a0, a1
; RV64ZBB-NEXT:  ret
%cmp = icmp ugt i32 %a, %b
%cond = select i1 %cmp, i32 %a, i32 %b
ret i32 %cond
}

define i64 @maxu_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: maxu_i64:
; RV64I:    # %bb.0:
; RV64I-NEXT:  bltu a1, a0, .LBB25_2
; RV64I-NEXT:  # %bb.1:
; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT:  .LBB25_2:
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: maxu_i64:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT:  maxu a0, a0, a1
; RV64ZBB-NEXT:  ret
%cmp = icmp ugt i64 %a, %b
%cond = select i1 %cmp, i64 %a, i64 %b
ret i64 %cond
}

declare i32 @llvm.abs.i32(i32, i1 immarg)

define i32 @abs_i32(i32 %x) {
; RV64I-LABEL: abs_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT:  sraiw a1, a0, 31
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  subw a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL:
abs_i32:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT:  sraiw a1, a0, 31
; RV64ZBB-NEXT:  xor a0, a0, a1
; RV64ZBB-NEXT:  subw a0, a0, a1
; RV64ZBB-NEXT:  ret
%abs = tail call i32 @llvm.abs.i32(i32 %x, i1 true)
ret i32 %abs
}

```

```

define signext i32 @abs_i32_sext(i32 signext %x) {
; RV64I-LABEL: abs_i32_sext:
; RV64I:      # %bb.0:
; RV64I-NEXT:  sraiw a1, a0, 31
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  subw a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: abs_i32_sext:
; RV64ZBB:     # %bb.0:
; RV64ZBB-NEXT:  sraiw a1, a0, 31
; RV64ZBB-NEXT:  xor a0, a0, a1
; RV64ZBB-NEXT:  subw a0, a0, a1
; RV64ZBB-NEXT:  ret
%abs = tail call i32 @llvm.abs.i32(i32 %x, i1 true)
ret i32 %abs
}

```

```

declare i64 @llvm.abs.i64(i64, i1 immarg)

```

```

define i64 @abs_i64(i64 %x) {
; RV64I-LABEL: abs_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srai a1, a0, 63
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  sub a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: abs_i64:
; RV64ZBB:
; RV64ZBB:     # %bb.0:
; RV64ZBB-NEXT:  neg a1, a0
; RV64ZBB-NEXT:  max a0, a0, a1
; RV64ZBB-NEXT:  ret
%abs = tail call i64 @llvm.abs.i64(i64 %x, i1 true)
ret i64 %abs
}

```

```

define i32 @zexth_i32(i32 %a) nounwind {
; RV64I-LABEL: zexth_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srli a0, a0, 48
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: zexth_i32:
; RV64ZBB:     # %bb.0:
; RV64ZBB-NEXT:  slli a0, a0, 48

```

```

; RV64ZBB-NEXT:  srli a0, a0, 48
; RV64ZBB-NEXT:  ret
%and = and i32 %a, 65535
ret i32 %and
}

define i64 @zexth_i64(i64 %a) nounwind {
; RV64I-LABEL: zexth_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srli a0, a0, 48
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: zexth_i64:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT:  zext.h a0, a0
; RV64ZBB-NEXT:  ret
%and = and i64 %a, 65535
ret i64 %and
}

declare i32 @llvm.bswap.i32(i32)

define signext i32 @bswap_i32(i32 signext %a) nounwind {
; RV64I-LABEL: bswap_i32:
;
; RV64I:      # %bb.0:
; RV64I-NEXT:  srliw a1, a0, 8
; RV64I-NEXT:  lui a2, 16
; RV64I-NEXT:  addiw a2, a2, -256
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  srliw a3, a0, 24
; RV64I-NEXT:  or a1, a1, a3
; RV64I-NEXT:  and a2, a0, a2
; RV64I-NEXT:  slliw a2, a2, 8
; RV64I-NEXT:  slliw a0, a0, 24
; RV64I-NEXT:  or a0, a0, a2
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: bswap_i32:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT:  rev8 a0, a0
; RV64ZBB-NEXT:  srai a0, a0, 32
; RV64ZBB-NEXT:  ret
%1 = tail call i32 @llvm.bswap.i32(i32 %a)
ret i32 %1
}

```

```

; Similar to bswap_i32 but the result is not sign extended.
define void @bswap_i32_nosext(i32 signext %a, ptr %x) nounwind {
; RV64I-LABEL: bswap_i32_nosext:
; RV64I:      # %bb.0:
; RV64I-NEXT: srliw a2, a0, 8
; RV64I-NEXT: lui a3, 16
; RV64I-NEXT: addi a3, a3, -256
; RV64I-NEXT: and a2, a2, a3
; RV64I-NEXT: srliw a4, a0, 24
; RV64I-NEXT: or a2, a2, a4
; RV64I-NEXT:
    and a3, a0, a3
; RV64I-NEXT: slli a3, a3, 8
; RV64I-NEXT: slli a0, a0, 24
; RV64I-NEXT: or a0, a0, a3
; RV64I-NEXT: or a0, a0, a2
; RV64I-NEXT: sw a0, 0(a1)
; RV64I-NEXT: ret
;
; RV64ZBB-LABEL: bswap_i32_nosext:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT: rev8 a0, a0
; RV64ZBB-NEXT: srli a0, a0, 32
; RV64ZBB-NEXT: sw a0, 0(a1)
; RV64ZBB-NEXT: ret
    %1 = tail call i32 @llvm.bswap.i32(i32 %a)
    store i32 %1, ptr %x
    ret void
}

```

```

declare i64 @llvm.bswap.i64(i64)

```

```

define i64 @bswap_i64(i64 %a) {
; RV64I-LABEL: bswap_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT: srli a1, a0, 40
; RV64I-NEXT: lui a2, 16
; RV64I-NEXT: addiw a2, a2, -256
; RV64I-NEXT: and a1, a1, a2
; RV64I-NEXT: srli a3, a0, 56
; RV64I-NEXT: or a1, a1, a3
; RV64I-NEXT: srli a3, a0, 24
; RV64I-NEXT: lui a4, 4080
; RV64I-NEXT: and a3, a3, a4
; RV64I-NEXT: srli a5, a0, 8
; RV64I-NEXT: srliw a5, a5, 24
; RV64I-NEXT: slli a5, a5, 24

```

```

; RV64I-NEXT:  or a3, a5, a3
; RV64I-NEXT:
    or a1, a3, a1
; RV64I-NEXT:  and a4, a0, a4
; RV64I-NEXT:  slli a4, a4, 24
; RV64I-NEXT:  srliw a3, a0, 24
; RV64I-NEXT:  slli a3, a3, 32
; RV64I-NEXT:  or a3, a4, a3
; RV64I-NEXT:  and a2, a0, a2
; RV64I-NEXT:  slli a2, a2, 40
; RV64I-NEXT:  slli a0, a0, 56
; RV64I-NEXT:  or a0, a0, a2
; RV64I-NEXT:  or a0, a0, a3
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: bswap_i64:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  rev8 a0, a0
; RV64ZBB-NEXT:  ret
    %1 = call i64 @llvm.bswap.i64(i64 %a)
    ret i64 %1
}
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -verify-machineinstrs < %s \
; RUN:  -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64I
; RUN: llc -mtriple=riscv64 -mattr=+zbkb -verify-machineinstrs < %s \
; RUN:  -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64ZBKB

```

```

define signext i32 @pack_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: pack_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srli a0, a0, 48
; RV64I-NEXT:  slliw a1, a1, 16
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBKB-LABEL: pack_i32:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  packw a0, a0, a1
; RV64ZBKB-NEXT:  ret
    %shl = and i32 %a, 65535
    %shl1 = shl i32 %b, 16
    %or = or i32 %shl1, %shl
    ret i32 %or
}

```

```

define signext i32 @pack_i32_2(i16 zeroext %a, i16 zeroext %b) nounwind {
; RV64I-LABEL: pack_i32_2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slliw
a1, a1, 16
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBKB-LABEL: pack_i32_2:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  packw a0, a0, a1
; RV64ZBKB-NEXT:  ret
%zexta = zext i16 %a to i32
%zextb = zext i16 %b to i32
%shl1 = shl i32 %zextb, 16
%or = or i32 %shl1, %zexta
ret i32 %or
}

; Test case where we don't have a sign_extend_inreg after the or.
define signext i32 @pack_i32_3(i16 zeroext %0, i16 zeroext %1, i32 signext %2) {
; RV64I-LABEL: pack_i32_3:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 16
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  addw a0, a0, a2
; RV64I-NEXT:  ret
;
; RV64ZBKB-LABEL: pack_i32_3:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  packw a0, a1, a0
; RV64ZBKB-NEXT:  addw a0, a0, a2
; RV64ZBKB-NEXT:  ret
%4 = zext i16 %0 to i32
%5 = shl nuw i32 %4, 16
%6 = zext i16 %1 to i32
%7 = or i32 %5, %6
%8 = add i32 %7, %2
ret i32 %8
}

define i64 @pack_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: pack_i64:
;
RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  slli a1, a1, 32

```

```

; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBKB-LABEL: pack_i64:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  pack a0, a0, a1
; RV64ZBKB-NEXT:  ret
%shl = and i64 %a, 4294967295
%shl1 = shl i64 %b, 32
%or = or i64 %shl1, %shl
ret i64 %or
}

```

```

define i64 @pack_i64_2(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: pack_i64_2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBKB-LABEL: pack_i64_2:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  pack a0, a0, a1
; RV64ZBKB-NEXT:  ret
%zexta = zext i32 %a to i64
%zextb = zext i32 %b to i64
%shl1 = shl i64 %zextb, 32
%or = or i64 %shl1, %zexta
ret i64 %or
}

```

```

define i64 @pack_i64_3(ptr %0, ptr %1) {
; RV64I-LABEL: pack_i64_3:
; RV64I:      #
%bb.0:
; RV64I-NEXT:  lw a0, 0(a0)
; RV64I-NEXT:  lwu a1, 0(a1)
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBKB-LABEL: pack_i64_3:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  lw a0, 0(a0)
; RV64ZBKB-NEXT:  lwu a1, 0(a1)
; RV64ZBKB-NEXT:  pack a0, a1, a0

```



```
; RV64ZBKB-NEXT:  ret
%3 = load i32, ptr %0, align 4
%4 = zext i32 %3 to i64
%5 = shl i64 %4, 32
%6 = load i32, ptr %1, align 4
%7 = zext i32 %6 to i64
%8 = or i64 %5, %7
ret i64 %8
}
```

```
define signext i32 @packh_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: packh_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  andi a0, a0, 255
; RV64I-NEXT:  slli a1, a1, 56
; RV64I-NEXT:  srli a1, a1, 48
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBKB-LABEL: packh_i32:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  packh a0, a0, a1
; RV64ZBKB-NEXT:  ret
%and = and i32 %a, 255
%and1 = shl i32 %b, 8
%shl = and i32 %and1, 65280
%or = or
i32 %shl, %and
ret i32 %or
}
```

```
define i32 @packh_i32_2(i32 %a, i32 %b) nounwind {
; RV64I-LABEL: packh_i32_2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  andi a0, a0, 255
; RV64I-NEXT:  andi a1, a1, 255
; RV64I-NEXT:  slliw a1, a1, 8
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBKB-LABEL: packh_i32_2:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  packh a0, a0, a1
; RV64ZBKB-NEXT:  ret
%and = and i32 %a, 255
%and1 = and i32 %b, 255
%shl = shl i32 %and1, 8
%or = or i32 %shl, %and
```

```
ret i32 %or
}
```

```
define i64 @packh_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: packh_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT: andi a0, a0, 255
; RV64I-NEXT: slli a1, a1, 56
; RV64I-NEXT: srli a1, a1, 48
; RV64I-NEXT: or a0, a1, a0
; RV64I-NEXT: ret
;
; RV64ZBKB-LABEL: packh_i64:
; RV64ZBKB:   # %bb.0:
; RV64ZBKB-NEXT: packh a0, a0, a1
; RV64ZBKB-NEXT: ret
%and = and i64 %a, 255
%and1 = shl i64 %b, 8
%shl = and i64 %and1, 65280
%or = or i64 %shl,
%and
ret i64 %or
}
```

```
define i64 @packh_i64_2(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: packh_i64_2:
; RV64I:      # %bb.0:
; RV64I-NEXT: andi a0, a0, 255
; RV64I-NEXT: andi a1, a1, 255
; RV64I-NEXT: slli a1, a1, 8
; RV64I-NEXT: or a0, a1, a0
; RV64I-NEXT: ret
;
; RV64ZBKB-LABEL: packh_i64_2:
; RV64ZBKB:   # %bb.0:
; RV64ZBKB-NEXT: packh a0, a0, a1
; RV64ZBKB-NEXT: ret
%and = and i64 %a, 255
%and1 = and i64 %b, 255
%shl = shl i64 %and1, 8
%or = or i64 %shl, %and
ret i64 %or
}
```

```
define zeroext i16 @packh_i16(i8 zeroext %a, i8 zeroext %b) nounwind {
; RV64I-LABEL: packh_i16:
; RV64I:      # %bb.0:
; RV64I-NEXT: slliw a1, a1, 8
```

```

; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBKB-LABEL: packh_i16:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  packh a0, a0, a1
; RV64ZBKB-NEXT:  ret
%zext = zext i8 %a to i16
%zext1 = zext i8 %b to i16
%shl = shl i16 %zext1, 8
%or = or i16 %shl, %zext
ret i16 %or
}

```

```

define zeroext i16 @packh_i16_2(i8
  zeroext %0, i8 zeroext %1, i8 zeroext %2) {
; RV64I-LABEL: packh_i16_2:
; RV64I:    # %bb.0:
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  slli a0, a0, 8
; RV64I-NEXT:  or a0, a0, a2
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srli a0, a0, 48
; RV64I-NEXT:  ret
;
; RV64ZBKB-LABEL: packh_i16_2:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  add a0, a1, a0
; RV64ZBKB-NEXT:  packh a0, a2, a0
; RV64ZBKB-NEXT:  ret
%4 = add i8 %1, %0
%5 = zext i8 %4 to i16
%6 = shl i16 %5, 8
%7 = zext i8 %2 to i16
%8 = or i16 %6, %7
ret i16 %8
}

```

```

define i64 @pack_i64_allWUsers(i32 signext %0, i32 signext %1, i32 signext %2) {
; RV64I-LABEL: pack_i64_allWUsers:
; RV64I:    # %bb.0:
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  slli a2, a2, 32
; RV64I-NEXT:  srli a2, a2, 32
; RV64I-NEXT:  or a0, a0, a2
; RV64I-NEXT:  ret
;

```

```

; RV64ZBKB-LABEL: pack_i64_allWUsers:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  add a0, a1, a0
; RV64ZBKB-NEXT:
    pack a0, a2, a0
; RV64ZBKB-NEXT:  ret
%4 = add i32 %1, %0
%5 = zext i32 %4 to i64
%6 = shl i64 %5, 32
%7 = zext i32 %2 to i64
%8 = or i64 %6, %7
ret i64 %8
}

define signext i32 @pack_i32_allWUsers(i16 zeroext %0, i16 zeroext %1, i16 zeroext %2) {
; RV64I-LABEL: pack_i32_allWUsers:
; RV64I:    # %bb.0:
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  slliw a0, a0, 16
; RV64I-NEXT:  or a0, a0, a2
; RV64I-NEXT:  ret
;
; RV64ZBKB-LABEL: pack_i32_allWUsers:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  add a0, a1, a0
; RV64ZBKB-NEXT:  packw a0, a2, a0
; RV64ZBKB-NEXT:  ret
%4 = add i16 %1, %0
%5 = zext i16 %4 to i32
%6 = shl i32 %5, 16
%7 = zext i16 %2 to i32
%8 = or i32 %6, %7
ret i32 %8
}

define i64 @pack_i64_imm() {
; RV64I-LABEL: pack_i64_imm:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 65793
; RV64I-NEXT:  addiw a0, a0, 16
; RV64I-NEXT:  slli a1, a0, 32
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBKB-LABEL:
    pack_i64_imm:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  lui a0, 65793

```

```

; RV64ZBKB-NEXT:  addi a0, a0, 16
; RV64ZBKB-NEXT:  pack a0, a0, a0
; RV64ZBKB-NEXT:  ret
ret i64 1157442765409226768 ; 0x0101010101010101
}
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -riscv-disable-using-constant-pool-for-large-ints -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefixes=RV64I,RV64-NOPOOL
; RUN: llc -mtriple=riscv64 -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefixes=RV64I,RV64I-POOL
; RUN: llc -mtriple=riscv64 -riscv-disable-using-constant-pool-for-large-ints -mattr=+zba \
; RUN: -riscv-experimental-rv64-legal-i32 -verify-machineinstrs < %s | FileCheck %s -check-prefix=RV64IZBA
; RUN: llc -mtriple=riscv64 -riscv-disable-using-constant-pool-for-large-ints -mattr=+zbb \
; RUN: -riscv-experimental-rv64-legal-i32 -verify-machineinstrs < %s | FileCheck %s -check-prefix=RV64IZBB
; RUN: llc -mtriple=riscv64 -riscv-disable-using-constant-pool-for-large-ints -mattr=+zbs \
; RUN: -riscv-experimental-rv64-legal-i32 -verify-machineinstrs < %s | FileCheck
% s -check-prefix=RV64IZBS
; RUN: llc -mtriple=riscv64 -riscv-disable-using-constant-pool-for-large-ints -mattr=+xtheadbb \
; RUN: -riscv-experimental-rv64-legal-i32 -verify-machineinstrs < %s | FileCheck %s -check-
prefix=RV64IXTHEADBB

; Materializing constants

; TODO: It would be preferable if anyext constant returns were sign rather
; than zero extended. See PR39092. For now, mark returns as explicitly signext
; (this matches what Clang would generate for equivalent C/C++ anyway).

define signext i32 @zero() nounwind {
; RV64I-LABEL: zero:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a0, 0
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: zero:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  li a0, 0
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: zero:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  li a0, 0
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: zero:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  li a0, 0
; RV64IZBS-NEXT:  ret
;

```

```

; RV64IXTHEADBB-LABEL: zero:
; RV64IXTHEADBB:    # %bb.0:
;
RV64IXTHEADBB-NEXT:  li a0, 0
; RV64IXTHEADBB-NEXT:  ret
ret i32 0
}

define signext i32 @pos_small() nounwind {
; RV64I-LABEL: pos_small:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a0, 2047
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: pos_small:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  li a0, 2047
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: pos_small:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  li a0, 2047
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: pos_small:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  li a0, 2047
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: pos_small:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  li a0, 2047
; RV64IXTHEADBB-NEXT:  ret
ret i32 2047
}

define signext i32 @neg_small() nounwind {
; RV64I-LABEL: neg_small:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a0, -2048
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: neg_small:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  li a0, -2048
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL:

```

```

neg_small:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  li a0, -2048
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: neg_small:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  li a0, -2048
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: neg_small:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  li a0, -2048
; RV64IXTHEADBB-NEXT:  ret
ret i32 -2048
}

```

```

define signext i32 @pos_i32() nounwind {
; RV64I-LABEL: pos_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 423811
; RV64I-NEXT:  addiw a0, a0, -1297
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: pos_i32:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 423811
; RV64IZBA-NEXT:  addiw a0, a0, -1297
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: pos_i32:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 423811
; RV64IZBB-NEXT:  addiw a0, a0, -1297
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: pos_i32:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 423811
; RV64IZBS-NEXT:  addiw a0, a0, -1297
; RV64IZBS-NEXT:
ret
;
; RV64IXTHEADBB-LABEL: pos_i32:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 423811
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -1297
; RV64IXTHEADBB-NEXT:  ret

```

```

ret i32 1735928559
}

define signext i32 @neg_i32() nounwind {
; RV64I-LABEL: neg_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 912092
; RV64I-NEXT:  addiw a0, a0, -273
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: neg_i32:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 912092
; RV64IZBA-NEXT:  addiw a0, a0, -273
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: neg_i32:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 912092
; RV64IZBB-NEXT:  addiw a0, a0, -273
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: neg_i32:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 912092
; RV64IZBS-NEXT:  addiw a0, a0, -273
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: neg_i32:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 912092
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -273
;
RV64IXTHEADBB-NEXT:  ret
ret i32 -559038737
}

define signext i32 @pos_i32_hi20_only() nounwind {
; RV64I-LABEL: pos_i32_hi20_only:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 16
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: pos_i32_hi20_only:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 16
; RV64IZBA-NEXT:  ret
;

```



```

; RV64IZBB-LABEL: pos_i32_hi20_only:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 16
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: pos_i32_hi20_only:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 16
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: pos_i32_hi20_only:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 16
; RV64IXTHEADBB-NEXT:  ret
ret i32 65536 ; 0x10000
}

```

```

define signext i32 @neg_i32_hi20_only() nounwind {
; RV64I-LABEL: neg_i32_hi20_only:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 1048560
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: neg_i32_hi20_only:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0,
1048560
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: neg_i32_hi20_only:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 1048560
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: neg_i32_hi20_only:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 1048560
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: neg_i32_hi20_only:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 1048560
; RV64IXTHEADBB-NEXT:  ret
ret i32 -65536 ; -0x10000
}

```

; This can be materialized with ADDI+SLLI, improving compressibility.

```

define signext i32 @imm_left_shifted_addi() nounwind {
; RV64I-LABEL: imm_left_shifted_addi:
; RV64I:      # %bb.0:
; RV64I-NEXT: lui a0, 32
; RV64I-NEXT: addiw a0, a0, -64
; RV64I-NEXT: ret
;
; RV64IZBA-LABEL: imm_left_shifted_addi:
; RV64IZBA:   # %bb.0:
; RV64IZBA-NEXT: lui a0, 32
; RV64IZBA-NEXT: addiw a0, a0, -64
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: imm_left_shifted_addi:
; RV64IZBB:   # %bb.0:
; RV64IZBB-NEXT: lui a0, 32
;
; RV64IZBB-NEXT: addiw a0, a0, -64
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: imm_left_shifted_addi:
; RV64IZBS:   # %bb.0:
; RV64IZBS-NEXT: lui a0, 32
; RV64IZBS-NEXT: addiw a0, a0, -64
; RV64IZBS-NEXT: ret
;
; RV64IXTHEADBB-LABEL: imm_left_shifted_addi:
; RV64IXTHEADBB:   # %bb.0:
; RV64IXTHEADBB-NEXT: lui a0, 32
; RV64IXTHEADBB-NEXT: addiw a0, a0, -64
; RV64IXTHEADBB-NEXT: ret
ret i32 131008 ; 0x1FFC0
}

```

; This can be materialized with ADDI+SRLI, improving compressibility.

```

define signext i32 @imm_right_shifted_addi() nounwind {
; RV64I-LABEL: imm_right_shifted_addi:
; RV64I:      # %bb.0:
; RV64I-NEXT: lui a0, 524288
; RV64I-NEXT: addiw a0, a0, -1
; RV64I-NEXT: ret
;
; RV64IZBA-LABEL: imm_right_shifted_addi:
; RV64IZBA:   # %bb.0:
; RV64IZBA-NEXT: lui a0, 524288
; RV64IZBA-NEXT: addiw a0, a0, -1
; RV64IZBA-NEXT: ret

```

```

;
; RV64IZBB-LABEL: imm_right_shifted_addi:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 524288
; RV64IZBB-NEXT:
;               addiw a0, a0, -1
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_right_shifted_addi:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 524288
; RV64IZBS-NEXT:  addiw a0, a0, -1
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_right_shifted_addi:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 524288
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -1
; RV64IXTHEADBB-NEXT:  ret
;               ret i32 2147483647 ; 0x7FFFFFFF
}

```

; This can be materialized with LUI+SRLI, improving compressibility.

```

define signext i32 @imm_right_shifted_lui() nounwind {
; RV64I-LABEL: imm_right_shifted_lui:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 56
; RV64I-NEXT:  addiw a0, a0, 580
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_right_shifted_lui:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 56
; RV64IZBA-NEXT:  addiw a0, a0, 580
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_right_shifted_lui:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 56
; RV64IZBB-NEXT:
;               addiw a0, a0, 580
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_right_shifted_lui:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 56
; RV64IZBS-NEXT:  addiw a0, a0, 580

```

```

; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_right_shifted_lui:
; RV64IXTHEADBB:   # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 56
; RV64IXTHEADBB-NEXT:  addiw a0, a0, 580
; RV64IXTHEADBB-NEXT:  ret
ret i32 229956 ; 0x38244
}

```

```

define i64 @imm64_1() nounwind {
; RV64I-LABEL: imm64_1:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a0, 1
; RV64I-NEXT:  slli a0, a0, 31
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm64_1:
; RV64IZBA:   # %bb.0:
; RV64IZBA-NEXT:  li a0, 1
; RV64IZBA-NEXT:  slli a0, a0, 31
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm64_1:
; RV64IZBB:   # %bb.0:
; RV64IZBB-NEXT:  li a0, 1
; RV64IZBB-NEXT:  slli a0, a0, 31
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm64_1:
; RV64IZBS:   # %bb.0:
; RV64IZBS-NEXT:  bseti a0, zero, 31
; RV64IZBS-NEXT:  ret
;
;
RV64IXTHEADBB-LABEL: imm64_1:
; RV64IXTHEADBB:   # %bb.0:
; RV64IXTHEADBB-NEXT:  li a0, 1
; RV64IXTHEADBB-NEXT:  slli a0, a0, 31
; RV64IXTHEADBB-NEXT:  ret
ret i64 2147483648 ; 0x8000_0000
}

```

```

define i64 @imm64_2() nounwind {
; RV64I-LABEL: imm64_2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a0, -1
; RV64I-NEXT:  srli a0, a0, 32

```

```

; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm64_2:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  li a0, -1
; RV64IZBA-NEXT:  srli a0, a0, 32
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm64_2:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  li a0, -1
; RV64IZBB-NEXT:  srli a0, a0, 32
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm64_2:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  li a0, -1
; RV64IZBS-NEXT:  srli a0, a0, 32
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm64_2:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  li a0, -1
; RV64IXTHEADBB-NEXT:  srli a0, a0, 32
; RV64IXTHEADBB-NEXT:  ret
ret i64 4294967295 ;
0xFFFF_FFFF
}

```

```

define i64 @imm64_3() nounwind {
; RV64I-LABEL: imm64_3:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a0, 1
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm64_3:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  li a0, 1
; RV64IZBA-NEXT:  slli a0, a0, 32
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm64_3:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  li a0, 1
; RV64IZBB-NEXT:  slli a0, a0, 32
; RV64IZBB-NEXT:  ret
;

```

```

; RV64IZBS-LABEL: imm64_3:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  bseti a0, zero, 32
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm64_3:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  li a0, 1
; RV64IXTHEADBB-NEXT:  slli a0, a0, 32
; RV64IXTHEADBB-NEXT:  ret
ret i64 4294967296 ; 0x1_0000_0000
}

define i64 @imm64_4() nounwind {
; RV64I-LABEL: imm64_4:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a0, -1
; RV64I-NEXT:  slli a0, a0, 63
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm64_4:
; RV64IZBA:
    # %bb.0:
; RV64IZBA-NEXT:  li a0, -1
; RV64IZBA-NEXT:  slli a0, a0, 63
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm64_4:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  li a0, -1
; RV64IZBB-NEXT:  slli a0, a0, 63
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm64_4:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  bseti a0, zero, 63
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm64_4:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  li a0, -1
; RV64IXTHEADBB-NEXT:  slli a0, a0, 63
; RV64IXTHEADBB-NEXT:  ret
ret i64 9223372036854775808 ; 0x8000_0000_0000_0000
}

define i64 @imm64_5() nounwind {
; RV64I-LABEL: imm64_5:

```

```

; RV64I:    # %bb.0:
; RV64I-NEXT:  li a0, -1
; RV64I-NEXT:  slli a0, a0, 63
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm64_5:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  li a0, -1
; RV64IZBA-NEXT:  slli a0, a0, 63
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm64_5:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  li a0, -1
; RV64IZBB-NEXT:
;               slli a0, a0, 63
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm64_5:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  bseti a0, zero, 63
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm64_5:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  li a0, -1
; RV64IXTHEADBB-NEXT:  slli a0, a0, 63
; RV64IXTHEADBB-NEXT:  ret
ret i64 -9223372036854775808 ; 0x8000_0000_0000_0000
}

```

```

define i64 @imm64_6() nounwind {
; RV64I-LABEL: imm64_6:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 9321
; RV64I-NEXT:  addi a0, a0, -1329
; RV64I-NEXT:  slli a0, a0, 35
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm64_6:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 9321
; RV64IZBA-NEXT:  addi a0, a0, -1329
; RV64IZBA-NEXT:  slli a0, a0, 35
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm64_6:
; RV64IZBB:    # %bb.0:

```

```

; RV64IZBB-NEXT:  lui a0, 9321
; RV64IZBB-NEXT:  addi a0, a0, -1329
; RV64IZBB-NEXT:  slli a0, a0, 35
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm64_6:
;
RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 9321
; RV64IZBS-NEXT:  addi a0, a0, -1329
; RV64IZBS-NEXT:  slli a0, a0, 35
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm64_6:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 9321
; RV64IXTHEADBB-NEXT:  addi a0, a0, -1329
; RV64IXTHEADBB-NEXT:  slli a0, a0, 35
; RV64IXTHEADBB-NEXT:  ret
ret i64 1311768464867721216 ; 0x1234_5678_0000_0000
}

```

```

define i64 @imm64_7() nounwind {
; RV64I-LABEL: imm64_7:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a0, 7
; RV64I-NEXT:  slli a0, a0, 36
; RV64I-NEXT:  addi a0, a0, 11
; RV64I-NEXT:  slli a0, a0, 24
; RV64I-NEXT:  addi a0, a0, 15
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm64_7:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  li a0, 7
; RV64IZBA-NEXT:  slli a0, a0, 36
; RV64IZBA-NEXT:  addi a0, a0, 11
; RV64IZBA-NEXT:  slli a0, a0, 24
; RV64IZBA-NEXT:  addi a0, a0, 15
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm64_7:
; RV64IZBB:    #
%bb.0:
; RV64IZBB-NEXT:  li a0, 7
; RV64IZBB-NEXT:  slli a0, a0, 36
; RV64IZBB-NEXT:  addi a0, a0, 11
; RV64IZBB-NEXT:  slli a0, a0, 24

```



```

; RV64IZBB-NEXT:  addi a0, a0, 15
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm64_7:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT:  li a0, 7
; RV64IZBS-NEXT:  slli a0, a0, 36
; RV64IZBS-NEXT:  addi a0, a0, 11
; RV64IZBS-NEXT:  slli a0, a0, 24
; RV64IZBS-NEXT:  addi a0, a0, 15
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm64_7:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  li a0, 7
; RV64IXTHEADBB-NEXT:  slli a0, a0, 36
; RV64IXTHEADBB-NEXT:  addi a0, a0, 11
; RV64IXTHEADBB-NEXT:  slli a0, a0, 24
; RV64IXTHEADBB-NEXT:  addi a0, a0, 15
; RV64IXTHEADBB-NEXT:  ret
ret i64 8070450532432478223 ; 0x7000_0000_0B00_000F
}

```

; TODO: it can be preferable to put constants that are expensive to materialise  
into the constant pool, especially for -Os.

```

define i64 @imm64_8() nounwind {
; RV64-NOPOOL-LABEL: imm64_8:
; RV64-NOPOOL:
;   # %bb.0:
; RV64-NOPOOL-NEXT:  lui a0, 583
; RV64-NOPOOL-NEXT:  addiw a0, a0, -1875
; RV64-NOPOOL-NEXT:  slli a0, a0, 14
; RV64-NOPOOL-NEXT:  addi a0, a0, -947
; RV64-NOPOOL-NEXT:  slli a0, a0, 12
; RV64-NOPOOL-NEXT:  addi a0, a0, 1511
; RV64-NOPOOL-NEXT:  slli a0, a0, 13
; RV64-NOPOOL-NEXT:  addi a0, a0, -272
; RV64-NOPOOL-NEXT:  ret
;
; RV64I-POOL-LABEL: imm64_8:
; RV64I-POOL:      # %bb.0:
; RV64I-POOL-NEXT:  lui a0, %hi(.LCPI17_0)
; RV64I-POOL-NEXT:  ld a0, %lo(.LCPI17_0)(a0)
; RV64I-POOL-NEXT:  ret
;
; RV64IZBA-LABEL: imm64_8:
; RV64IZBA:      # %bb.0:
; RV64IZBA-NEXT:  lui a0, 596523

```

```

; RV64IZBA-NEXT:  addi a0, a0, 965
; RV64IZBA-NEXT:  slli.uw a0, a0, 13
; RV64IZBA-NEXT:  addi a0, a0, -1347
; RV64IZBA-NEXT:  slli a0, a0, 12
; RV64IZBA-NEXT:  addi a0, a0, -529
; RV64IZBA-NEXT:  slli a0, a0, 4
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm64_8:
; RV64IZBB:      # %bb.0:
; RV64IZBB-NEXT:  lui a0, 583
; RV64IZBB-NEXT:  addiw
a0, a0, -1875
; RV64IZBB-NEXT:  slli a0, a0, 14
; RV64IZBB-NEXT:  addi a0, a0, -947
; RV64IZBB-NEXT:  slli a0, a0, 12
; RV64IZBB-NEXT:  addi a0, a0, 1511
; RV64IZBB-NEXT:  slli a0, a0, 13
; RV64IZBB-NEXT:  addi a0, a0, -272
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm64_8:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT:  lui a0, 583
; RV64IZBS-NEXT:  addiw a0, a0, -1875
; RV64IZBS-NEXT:  slli a0, a0, 14
; RV64IZBS-NEXT:  addi a0, a0, -947
; RV64IZBS-NEXT:  slli a0, a0, 12
; RV64IZBS-NEXT:  addi a0, a0, 1511
; RV64IZBS-NEXT:  slli a0, a0, 13
; RV64IZBS-NEXT:  addi a0, a0, -272
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm64_8:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 583
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -1875
; RV64IXTHEADBB-NEXT:  slli a0, a0, 14
; RV64IXTHEADBB-NEXT:  addi a0, a0, -947
; RV64IXTHEADBB-NEXT:  slli a0, a0, 12
; RV64IXTHEADBB-NEXT:  addi a0, a0, 1511
; RV64IXTHEADBB-NEXT:  slli a0, a0, 13
; RV64IXTHEADBB-NEXT:
    addi a0, a0, -272
; RV64IXTHEADBB-NEXT:  ret
ret i64 1311768467463790320 ; 0x1234_5678_9ABC_DEF0
}

```

```

define i64 @imm64_9() nounwind {
; RV64I-LABEL: imm64_9:
; RV64I:      # %bb.0:
; RV64I-NEXT: li a0, -1
; RV64I-NEXT: ret
;
; RV64IZBA-LABEL: imm64_9:
; RV64IZBA:   # %bb.0:
; RV64IZBA-NEXT: li a0, -1
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: imm64_9:
; RV64IZBB:   # %bb.0:
; RV64IZBB-NEXT: li a0, -1
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: imm64_9:
; RV64IZBS:   # %bb.0:
; RV64IZBS-NEXT: li a0, -1
; RV64IZBS-NEXT: ret
;
; RV64IXTHEADBB-LABEL: imm64_9:
; RV64IXTHEADBB:   # %bb.0:
; RV64IXTHEADBB-NEXT: li a0, -1
; RV64IXTHEADBB-NEXT: ret
ret i64 -1
}

```

; Various cases where extraneous ADDIs can be inserted where a (left shifted)  
; LUI suffices.

```

define i64 @imm_left_shifted_lui_1() nounwind {
; RV64I-LABEL: imm_left_shifted_lui_1:
; RV64I:      # %bb.0:
; RV64I-NEXT: lui a0, 262145
; RV64I-NEXT: slli a0, a0, 1
;
; RV64I-NEXT: ret
;
; RV64IZBA-LABEL: imm_left_shifted_lui_1:
; RV64IZBA:   # %bb.0:
; RV64IZBA-NEXT: lui a0, 262145
; RV64IZBA-NEXT: slli a0, a0, 1
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: imm_left_shifted_lui_1:
; RV64IZBB:   # %bb.0:
; RV64IZBB-NEXT: lui a0, 262145

```

```

; RV64IZBB-NEXT:  slli a0, a0, 1
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_left_shifted_lui_1:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT:  lui a0, 262145
; RV64IZBS-NEXT:  slli a0, a0, 1
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_left_shifted_lui_1:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 262145
; RV64IXTHEADBB-NEXT:  slli a0, a0, 1
; RV64IXTHEADBB-NEXT:  ret
ret i64 2147491840 ; 0x8000_2000
}

```

```

define i64 @imm_left_shifted_lui_2() nounwind {
; RV64I-LABEL: imm_left_shifted_lui_2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a0, 262145
; RV64I-NEXT:  slli a0, a0, 2
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_left_shifted_lui_2:
; RV64IZBA:
# %bb.0:
; RV64IZBA-NEXT:  lui a0, 262145
; RV64IZBA-NEXT:  slli a0, a0, 2
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_left_shifted_lui_2:
; RV64IZBB:      # %bb.0:
; RV64IZBB-NEXT:  lui a0, 262145
; RV64IZBB-NEXT:  slli a0, a0, 2
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_left_shifted_lui_2:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT:  lui a0, 262145
; RV64IZBS-NEXT:  slli a0, a0, 2
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_left_shifted_lui_2:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 262145
; RV64IXTHEADBB-NEXT:  slli a0, a0, 2
; RV64IXTHEADBB-NEXT:  ret
}

```

```
ret i64 4294983680 ; 0x1_0000_4000
}
```

```
define i64 @imm_left_shifted_lui_3() nounwind {
; RV64I-LABEL: imm_left_shifted_lui_3:
; RV64I:      # %bb.0:
; RV64I-NEXT: lui a0, 4097
; RV64I-NEXT: slli a0, a0, 20
; RV64I-NEXT: ret
;
; RV64IZBA-LABEL: imm_left_shifted_lui_3:
; RV64IZBA:   # %bb.0:
; RV64IZBA-NEXT: lui a0, 4097
; RV64IZBA-NEXT: slli a0, a0, 20
;
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: imm_left_shifted_lui_3:
; RV64IZBB:   # %bb.0:
; RV64IZBB-NEXT: lui a0, 4097
; RV64IZBB-NEXT: slli a0, a0, 20
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: imm_left_shifted_lui_3:
; RV64IZBS:   # %bb.0:
; RV64IZBS-NEXT: lui a0, 4097
; RV64IZBS-NEXT: slli a0, a0, 20
; RV64IZBS-NEXT: ret
;
; RV64IXTHEADBB-LABEL: imm_left_shifted_lui_3:
; RV64IXTHEADBB:   # %bb.0:
; RV64IXTHEADBB-NEXT: lui a0, 4097
; RV64IXTHEADBB-NEXT: slli a0, a0, 20
; RV64IXTHEADBB-NEXT: ret
ret i64 17596481011712 ; 0x1001_0000_0000
}
```

; Various cases where extraneous ADDIs can be inserted where a (right shifted)  
; LUI suffices, or where multiple ADDIs can be used instead of a single LUI.

```
define i64 @imm_right_shifted_lui_1() nounwind {
; RV64I-LABEL: imm_right_shifted_lui_1:
; RV64I:      # %bb.0:
; RV64I-NEXT: lui a0, 983056
; RV64I-NEXT: srli a0, a0, 16
; RV64I-NEXT: ret
;
; RV64IZBA-LABEL: imm_right_shifted_lui_1:
```

```

; RV64IZBA:
    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 983056
; RV64IZBA-NEXT:  srli a0, a0, 16
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_right_shifted_lui_1:
; RV64IZBB:      # %bb.0:
; RV64IZBB-NEXT:  lui a0, 983056
; RV64IZBB-NEXT:  srli a0, a0, 16
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_right_shifted_lui_1:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT:  lui a0, 983056
; RV64IZBS-NEXT:  srli a0, a0, 16
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_right_shifted_lui_1:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 983056
; RV64IXTHEADBB-NEXT:  srli a0, a0, 16
; RV64IXTHEADBB-NEXT:  ret
ret i64 281474976706561 ; 0xFFFF_FFFF_F001
}

```

```

define i64 @imm_right_shifted_lui_2() nounwind {
; RV64I-LABEL: imm_right_shifted_lui_2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a0, 1044481
; RV64I-NEXT:  slli a0, a0, 12
; RV64I-NEXT:  srli a0, a0, 24
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_right_shifted_lui_2:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:
    lui a0, 1044481
; RV64IZBA-NEXT:  slli a0, a0, 12
; RV64IZBA-NEXT:  srli a0, a0, 24
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_right_shifted_lui_2:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 1044481
; RV64IZBB-NEXT:  slli a0, a0, 12
; RV64IZBB-NEXT:  srli a0, a0, 24
; RV64IZBB-NEXT:  ret
}

```

```

;
; RV64IZBS-LABEL: imm_right_shifted_lui_2:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 1044481
; RV64IZBS-NEXT:  slli a0, a0, 12
; RV64IZBS-NEXT:  srli a0, a0, 24
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_right_shifted_lui_2:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 1044481
; RV64IXTHEADBB-NEXT:  slli a0, a0, 12
; RV64IXTHEADBB-NEXT:  srli a0, a0, 24
; RV64IXTHEADBB-NEXT:  ret
ret i64 1099511623681 ; 0xFF_FFFF_F001
}

```

; We can materialize the upper bits with a single (shifted) LUI, but that option  
; can be missed due to the lower bits, which aren't just 1s or just 0s.

```

define i64 @imm_decoupled_lui_addi() nounwind
{
; RV64I-LABEL: imm_decoupled_lui_addi:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 4097
; RV64I-NEXT:  slli a0, a0, 20
; RV64I-NEXT:  addi a0, a0, -3
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_decoupled_lui_addi:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 4097
; RV64IZBA-NEXT:  slli a0, a0, 20
; RV64IZBA-NEXT:  addi a0, a0, -3
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_decoupled_lui_addi:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 4097
; RV64IZBB-NEXT:  slli a0, a0, 20
; RV64IZBB-NEXT:  addi a0, a0, -3
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_decoupled_lui_addi:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 4097
; RV64IZBS-NEXT:  slli a0, a0, 20
; RV64IZBS-NEXT:  addi a0, a0, -3

```

```

; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_decoupled_lui_addi:
; RV64IXTHEADBB:   # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 4097
; RV64IXTHEADBB-NEXT:  slli a0, a0, 20
; RV64IXTHEADBB-NEXT:  addi a0, a0, -3
; RV64IXTHEADBB-NEXT:
    ret
ret i64 17596481011709 ; 0x1000_FFFF_FFFD
}

```

; This constant can be materialized for RV64 with LUI+SRLI+XORI.

```

define i64 @imm_end_xori_1() nounwind {
; RV64I-LABEL: imm_end_xori_1:
; RV64I:       # %bb.0:
; RV64I-NEXT:  lui a0, 983040
; RV64I-NEXT:  srli a0, a0, 3
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_end_xori_1:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 983040
; RV64IZBA-NEXT:  srli a0, a0, 3
; RV64IZBA-NEXT:  not a0, a0
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_end_xori_1:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 983040
; RV64IZBB-NEXT:  srli a0, a0, 3
; RV64IZBB-NEXT:  not a0, a0
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_end_xori_1:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 983040
; RV64IZBS-NEXT:  srli a0, a0, 3
; RV64IZBS-NEXT:  not a0, a0
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_end_xori_1:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:
    lui a0, 983040
; RV64IXTHEADBB-NEXT:  srli a0, a0, 3

```



```
; RV64IXTHEADBB-NEXT:  not a0, a0
; RV64IXTHEADBB-NEXT:  ret
ret i64 -2305843009180139521 ; 0xE000_0000_01FF_FFFF
}
```

; This constant can be materialized for RV64 with ADDI+SLLI+ADDI+ADDI.

```
define i64 @imm_end_2addi_1() nounwind {
; RV64I-LABEL: imm_end_2addi_1:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a0, -2047
; RV64I-NEXT:  slli a0, a0, 39
; RV64I-NEXT:  addi a0, a0, -2048
; RV64I-NEXT:  addi a0, a0, -1
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_end_2addi_1:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  li a0, -2047
; RV64IZBA-NEXT:  slli a0, a0, 39
; RV64IZBA-NEXT:  addi a0, a0, -2048
; RV64IZBA-NEXT:  addi a0, a0, -1
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_end_2addi_1:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  li a0, -2047
; RV64IZBB-NEXT:  slli a0, a0, 39
; RV64IZBB-NEXT:  addi a0, a0, -2048
; RV64IZBB-NEXT:  addi a0, a0, -1
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_end_2addi_1:
;
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  li a0, -2047
; RV64IZBS-NEXT:  slli a0, a0, 39
; RV64IZBS-NEXT:  addi a0, a0, -2048
; RV64IZBS-NEXT:  addi a0, a0, -1
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_end_2addi_1:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  li a0, -2047
; RV64IXTHEADBB-NEXT:  slli a0, a0, 39
; RV64IXTHEADBB-NEXT:  addi a0, a0, -2048
; RV64IXTHEADBB-NEXT:  addi a0, a0, -1
; RV64IXTHEADBB-NEXT:  ret
}
```

```
ret i64 -1125350151030785 ; 0xFFFC_007F_FFFF_F7FF
}
```

; This constant can be more efficiently materialized for RV64 if we use two  
; registers instead of one.

```
define i64 @imm_2reg_1() nounwind {
; RV64I-LABEL: imm_2reg_1:
; RV64I:      # %bb.0:
; RV64I-NEXT: lui a0, 74565
; RV64I-NEXT: addiw a0, a0, 1656
; RV64I-NEXT: slli a1, a0, 57
; RV64I-NEXT: add a0, a0, a1
; RV64I-NEXT: ret
;
; RV64IZBA-LABEL: imm_2reg_1:
; RV64IZBA:   # %bb.0:
; RV64IZBA-NEXT: lui a0, 74565
; RV64IZBA-NEXT: addiw a0, a0, 1656
; RV64IZBA-NEXT:
;               slli a1, a0, 57
; RV64IZBA-NEXT: add a0, a0, a1
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: imm_2reg_1:
; RV64IZBB:   # %bb.0:
; RV64IZBB-NEXT: lui a0, 74565
; RV64IZBB-NEXT: addiw a0, a0, 1656
; RV64IZBB-NEXT: slli a1, a0, 57
; RV64IZBB-NEXT: add a0, a0, a1
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: imm_2reg_1:
; RV64IZBS:   # %bb.0:
; RV64IZBS-NEXT: lui a0, 74565
; RV64IZBS-NEXT: addiw a0, a0, 1656
; RV64IZBS-NEXT: slli a1, a0, 57
; RV64IZBS-NEXT: add a0, a0, a1
; RV64IZBS-NEXT: ret
;
; RV64IXTHEADBB-LABEL: imm_2reg_1:
; RV64IXTHEADBB:   # %bb.0:
; RV64IXTHEADBB-NEXT: lui a0, 74565
; RV64IXTHEADBB-NEXT: addiw a0, a0, 1656
; RV64IXTHEADBB-NEXT: slli a1, a0, 57
; RV64IXTHEADBB-NEXT: add a0, a0, a1
; RV64IXTHEADBB-NEXT: ret
ret i64 -1152921504301427080 ; 0xF000_0000_1234_5678
}
```

```

}

; FIXME: This should use a single ADDI for the immediate.
define void @imm_store_i16_neg1(ptr %p) nounwind {
; RV64I-LABEL: imm_store_i16_neg1:
;
RV64I:    # %bb.0:
; RV64I-NEXT:  li a1, -1
; RV64I-NEXT:  sh a1, 0(a0)
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_store_i16_neg1:
RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  li a1, -1
; RV64IZBA-NEXT:  sh a1, 0(a0)
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_store_i16_neg1:
RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  li a1, -1
; RV64IZBB-NEXT:  sh a1, 0(a0)
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_store_i16_neg1:
RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  li a1, -1
; RV64IZBS-NEXT:  sh a1, 0(a0)
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_store_i16_neg1:
RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  li a1, -1
; RV64IXTHEADBB-NEXT:  sh a1, 0(a0)
; RV64IXTHEADBB-NEXT:  ret
store i16 -1, ptr %p
ret void
}

```

```

; FIXME: This should use a single ADDI for the immediate.
define void @imm_store_i32_neg1(ptr %p) nounwind {
; RV64I-LABEL: imm_store_i32_neg1:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a1, -1
; RV64I-NEXT:
sw a1, 0(a0)
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_store_i32_neg1:

```

```

; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  li a1, -1
; RV64IZBA-NEXT:  sw a1, 0(a0)
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_store_i32_neg1:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  li a1, -1
; RV64IZBB-NEXT:  sw a1, 0(a0)
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_store_i32_neg1:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  li a1, -1
; RV64IZBS-NEXT:  sw a1, 0(a0)
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_store_i32_neg1:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  li a1, -1
; RV64IXTHEADBB-NEXT:  sw a1, 0(a0)
; RV64IXTHEADBB-NEXT:  ret
store i32 -1, ptr %p
ret void
}

```

```

define i64 @imm_5372288229() {
; RV64I-LABEL: imm_5372288229:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 160
; RV64I-NEXT:  addiw a0, a0, 437
; RV64I-NEXT:  slli a0, a0, 13
; RV64I-NEXT:  addi a0, a0, -795
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_5372288229:
; RV64IZBA:
;   # %bb.0:
; RV64IZBA-NEXT:  lui a0, 655797
; RV64IZBA-NEXT:  slli.uw a0, a0, 1
; RV64IZBA-NEXT:  addi a0, a0, -795
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_5372288229:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 160
; RV64IZBB-NEXT:  addiw a0, a0, 437
; RV64IZBB-NEXT:  slli a0, a0, 13

```

```

; RV64IZBB-NEXT:  addi a0, a0, -795
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_5372288229:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT:  lui a0, 263018
; RV64IZBS-NEXT:  addiw a0, a0, -795
; RV64IZBS-NEXT:  bseti a0, a0, 32
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_5372288229:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 160
; RV64IXTHEADBB-NEXT:  addiw a0, a0, 437
; RV64IXTHEADBB-NEXT:  slli a0, a0, 13
; RV64IXTHEADBB-NEXT:  addi a0, a0, -795
; RV64IXTHEADBB-NEXT:  ret
ret i64 5372288229
}

```

```

define i64 @imm_neg_5372288229() {
; RV64I-LABEL: imm_neg_5372288229:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a0, 1048416
; RV64I-NEXT:
    addiw a0, a0, -437
; RV64I-NEXT:  slli a0, a0, 13
; RV64I-NEXT:  addi a0, a0, 795
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_neg_5372288229:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 611378
; RV64IZBA-NEXT:  addiw a0, a0, 265
; RV64IZBA-NEXT:  sh1add a0, a0, a0
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_neg_5372288229:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 1048416
; RV64IZBB-NEXT:  addiw a0, a0, -437
; RV64IZBB-NEXT:  slli a0, a0, 13
; RV64IZBB-NEXT:  addi a0, a0, 795
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_neg_5372288229:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 785558

```

```

; RV64IZBS-NEXT:   addiw a0, a0, 795
; RV64IZBS-NEXT:   bclri a0, a0, 32
; RV64IZBS-NEXT:   ret
;
; RV64IXTHEADBB-LABEL: imm_neg_5372288229:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:   lui a0, 1048416
; RV64IXTHEADBB-NEXT:   addiw a0, a0, -437
; RV64IXTHEADBB-NEXT:   slli a0, a0, 13
; RV64IXTHEADBB-NEXT:   addi a0, a0,
795
; RV64IXTHEADBB-NEXT:   ret
ret i64 -5372288229
}

```

```

define i64 @imm_8953813715() {
; RV64I-LABEL: imm_8953813715:
; RV64I:    # %bb.0:
; RV64I-NEXT:   lui a0, 267
; RV64I-NEXT:   addiw a0, a0, -637
; RV64I-NEXT:   slli a0, a0, 13
; RV64I-NEXT:   addi a0, a0, -1325
; RV64I-NEXT:   ret
;
; RV64IZBA-LABEL: imm_8953813715:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:   lui a0, 437198
; RV64IZBA-NEXT:   addiw a0, a0, -265
; RV64IZBA-NEXT:   sh2add a0, a0, a0
; RV64IZBA-NEXT:   ret
;
; RV64IZBB-LABEL: imm_8953813715:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:   lui a0, 267
; RV64IZBB-NEXT:   addiw a0, a0, -637
; RV64IZBB-NEXT:   slli a0, a0, 13
; RV64IZBB-NEXT:   addi a0, a0, -1325
; RV64IZBB-NEXT:   ret
;
; RV64IZBS-LABEL: imm_8953813715:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:   lui a0, 88838
; RV64IZBS-NEXT:   addiw a0, a0, -1325
; RV64IZBS-NEXT:   bseti a0, a0, 33
; RV64IZBS-NEXT:   ret
;
; RV64IXTHEADBB-LABEL: imm_8953813715:
; RV64IXTHEADBB:    # %bb.0:

```

```

;
RV64IXTHEADBB-NEXT:  lui a0, 267
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -637
; RV64IXTHEADBB-NEXT:  slli a0, a0, 13
; RV64IXTHEADBB-NEXT:  addi a0, a0, -1325
; RV64IXTHEADBB-NEXT:  ret
ret i64 8953813715
}

define i64 @imm_neg_8953813715() {
; RV64I-LABEL: imm_neg_8953813715:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a0, 1048309
; RV64I-NEXT:  addiw a0, a0, 637
; RV64I-NEXT:  slli a0, a0, 13
; RV64I-NEXT:  addi a0, a0, 1325
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_neg_8953813715:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 611378
; RV64IZBA-NEXT:  addiw a0, a0, 265
; RV64IZBA-NEXT:  sh2add a0, a0, a0
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_neg_8953813715:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 1048309
; RV64IZBB-NEXT:  addiw a0, a0, 637
; RV64IZBB-NEXT:  slli a0, a0, 13
; RV64IZBB-NEXT:  addi a0, a0, 1325
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_neg_8953813715:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:
    lui a0, 959738
; RV64IZBS-NEXT:  addiw a0, a0, 1325
; RV64IZBS-NEXT:  bclri a0, a0, 33
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_neg_8953813715:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 1048309
; RV64IXTHEADBB-NEXT:  addiw a0, a0, 637
; RV64IXTHEADBB-NEXT:  slli a0, a0, 13
; RV64IXTHEADBB-NEXT:  addi a0, a0, 1325
; RV64IXTHEADBB-NEXT:  ret

```

```

ret i64 -8953813715
}

define i64 @imm_16116864687() {
; RV64I-LABEL: imm_16116864687:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 961
; RV64I-NEXT:  addiw a0, a0, -1475
; RV64I-NEXT:  slli a0, a0, 12
; RV64I-NEXT:  addi a0, a0, 1711
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_16116864687:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 437198
; RV64IZBA-NEXT:  addiw a0, a0, -265
; RV64IZBA-NEXT:  sh3add a0, a0, a0
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_16116864687:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 961
; RV64IZBB-NEXT:  addiw a0, a0, -1475
; RV64IZBB-NEXT:
;               slli a0, a0, 12
; RV64IZBB-NEXT:  addi a0, a0, 1711
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_16116864687:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 961
; RV64IZBS-NEXT:  addiw a0, a0, -1475
; RV64IZBS-NEXT:  slli a0, a0, 12
; RV64IZBS-NEXT:  addi a0, a0, 1711
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_16116864687:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 961
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -1475
; RV64IXTHEADBB-NEXT:  slli a0, a0, 12
; RV64IXTHEADBB-NEXT:  addi a0, a0, 1711
; RV64IXTHEADBB-NEXT:  ret
ret i64 16116864687
}

define i64 @imm_neg_16116864687() {
; RV64I-LABEL: imm_neg_16116864687:

```



```

; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 1047615
; RV64I-NEXT:  addiw a0, a0, 1475
; RV64I-NEXT:  slli a0, a0, 12
; RV64I-NEXT:  addi a0, a0, -1711
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_neg_16116864687:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 611378
; RV64IZBA-NEXT:
addiw a0, a0, 265
; RV64IZBA-NEXT:  sh3add a0, a0, a0
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_neg_16116864687:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 1047615
; RV64IZBB-NEXT:  addiw a0, a0, 1475
; RV64IZBB-NEXT:  slli a0, a0, 12
; RV64IZBB-NEXT:  addi a0, a0, -1711
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_neg_16116864687:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 1047615
; RV64IZBS-NEXT:  addiw a0, a0, 1475
; RV64IZBS-NEXT:  slli a0, a0, 12
; RV64IZBS-NEXT:  addi a0, a0, -1711
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_neg_16116864687:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 1047615
; RV64IXTHEADBB-NEXT:  addiw a0, a0, 1475
; RV64IXTHEADBB-NEXT:  slli a0, a0, 12
; RV64IXTHEADBB-NEXT:  addi a0, a0, -1711
; RV64IXTHEADBB-NEXT:  ret
ret i64 -16116864687
}

define i64 @imm_2344336315() {
; RV64I-LABEL: imm_2344336315:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 143087
; RV64I-NEXT:
slli a0, a0, 2
; RV64I-NEXT:  addi a0, a0, -1093

```

```

; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_2344336315:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 143087
; RV64IZBA-NEXT:  slli a0, a0, 2
; RV64IZBA-NEXT:  addi a0, a0, -1093
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_2344336315:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 143087
; RV64IZBB-NEXT:  slli a0, a0, 2
; RV64IZBB-NEXT:  addi a0, a0, -1093
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_2344336315:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 143087
; RV64IZBS-NEXT:  slli a0, a0, 2
; RV64IZBS-NEXT:  addi a0, a0, -1093
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_2344336315:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 143087
; RV64IXTHEADBB-NEXT:  slli a0, a0, 2
; RV64IXTHEADBB-NEXT:  addi a0, a0, -1093
; RV64IXTHEADBB-NEXT:  ret
ret i64 2344336315 ; 0x8bbbbbbb
}

define i64 @imm_70370820078523() {
; RV64-NOPOOL-LABEL: imm_70370820078523:
;
RV64-NOPOOL:    # %bb.0:
; RV64-NOPOOL-NEXT:  lui a0, 256
; RV64-NOPOOL-NEXT:  addiw a0, a0, 31
; RV64-NOPOOL-NEXT:  slli a0, a0, 12
; RV64-NOPOOL-NEXT:  addi a0, a0, -273
; RV64-NOPOOL-NEXT:  slli a0, a0, 14
; RV64-NOPOOL-NEXT:  addi a0, a0, -1093
; RV64-NOPOOL-NEXT:  ret
;
; RV64I-POOL-LABEL: imm_70370820078523:
; RV64I-POOL:    # %bb.0:
; RV64I-POOL-NEXT:  lui a0, %hi(.LCPI37_0)
; RV64I-POOL-NEXT:  ld a0, %lo(.LCPI37_0)(a0)

```

```

; RV64I-POOL-NEXT:  ret
;
; RV64IZBA-LABEL: imm_70370820078523:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 256
; RV64IZBA-NEXT:  addiw a0, a0, 31
; RV64IZBA-NEXT:  slli a0, a0, 12
; RV64IZBA-NEXT:  addi a0, a0, -273
; RV64IZBA-NEXT:  slli a0, a0, 14
; RV64IZBA-NEXT:  addi a0, a0, -1093
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_70370820078523:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 256
; RV64IZBB-NEXT:  addiw a0, a0, 31
; RV64IZBB-NEXT:  slli a0, a0, 12
; RV64IZBB-NEXT:  addi a0, a0, -273
;
RV64IZBB-NEXT:  slli a0, a0, 14
; RV64IZBB-NEXT:  addi a0, a0, -1093
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_70370820078523:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 506812
; RV64IZBS-NEXT:  addiw a0, a0, -1093
; RV64IZBS-NEXT:  bseti a0, a0, 46
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_70370820078523:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 256
; RV64IXTHEADBB-NEXT:  addiw a0, a0, 31
; RV64IXTHEADBB-NEXT:  slli a0, a0, 12
; RV64IXTHEADBB-NEXT:  addi a0, a0, -273
; RV64IXTHEADBB-NEXT:  slli a0, a0, 14
; RV64IXTHEADBB-NEXT:  addi a0, a0, -1093
; RV64IXTHEADBB-NEXT:  ret
ret i64 70370820078523 ; 0x40007bbbbbbb
}

```

```

define i64 @imm_neg_9223372034778874949() {
; RV64I-LABEL: imm_neg_9223372034778874949:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 506812
; RV64I-NEXT:  addiw a0, a0, -1093
; RV64I-NEXT:  slli a1, a0, 63

```

```

; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_neg_9223372034778874949:
;
RV64IZBA:      # %bb.0:
; RV64IZBA-NEXT:  lui a0, 506812
; RV64IZBA-NEXT:  addiw a0, a0, -1093
; RV64IZBA-NEXT:  slli a1, a0, 63
; RV64IZBA-NEXT:  add a0, a0, a1
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_neg_9223372034778874949:
; RV64IZBB:      # %bb.0:
; RV64IZBB-NEXT:  lui a0, 506812
; RV64IZBB-NEXT:  addiw a0, a0, -1093
; RV64IZBB-NEXT:  slli a1, a0, 63
; RV64IZBB-NEXT:  add a0, a0, a1
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_neg_9223372034778874949:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT:  lui a0, 506812
; RV64IZBS-NEXT:  addiw a0, a0, -1093
; RV64IZBS-NEXT:  bseti a0, a0, 63
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_neg_9223372034778874949:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 506812
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -1093
; RV64IXTHEADBB-NEXT:  slli a1, a0, 63
; RV64IXTHEADBB-NEXT:  add a0, a0, a1
; RV64IXTHEADBB-NEXT:  ret
ret i64 -9223372034778874949 ; 0x8000000007bbbbbbb
}

```

```

define
i64 @imm_neg_9223301666034697285() {
; RV64-NOPOOL-LABEL: imm_neg_9223301666034697285:
; RV64-NOPOOL:      # %bb.0:
; RV64-NOPOOL-NEXT:  lui a0, 917505
; RV64-NOPOOL-NEXT:  slli a0, a0, 8
; RV64-NOPOOL-NEXT:  addi a0, a0, 31
; RV64-NOPOOL-NEXT:  slli a0, a0, 12
; RV64-NOPOOL-NEXT:  addi a0, a0, -273
; RV64-NOPOOL-NEXT:  slli a0, a0, 14
; RV64-NOPOOL-NEXT:  addi a0, a0, -1093

```

```

; RV64-NOPOOL-NEXT:  ret
;
; RV64I-POOL-LABEL: imm_neg_9223301666034697285:
; RV64I-POOL:        # %bb.0:
; RV64I-POOL-NEXT:   lui a0, %hi(.LCPI39_0)
; RV64I-POOL-NEXT:   ld a0, %lo(.LCPI39_0)(a0)
; RV64I-POOL-NEXT:   ret
;
; RV64IZBA-LABEL: imm_neg_9223301666034697285:
; RV64IZBA:         # %bb.0:
; RV64IZBA-NEXT:    lui a0, 917505
; RV64IZBA-NEXT:    slli a0, a0, 8
; RV64IZBA-NEXT:    addi a0, a0, 31
; RV64IZBA-NEXT:    slli a0, a0, 12
; RV64IZBA-NEXT:    addi a0, a0, -273
; RV64IZBA-NEXT:    slli a0, a0, 14
; RV64IZBA-NEXT:    addi a0, a0, -1093
; RV64IZBA-NEXT:    ret
;
; RV64IZBB-LABEL:
imm_neg_9223301666034697285:
; RV64IZBB:         # %bb.0:
; RV64IZBB-NEXT:    lui a0, 917505
; RV64IZBB-NEXT:    slli a0, a0, 8
; RV64IZBB-NEXT:    addi a0, a0, 31
; RV64IZBB-NEXT:    slli a0, a0, 12
; RV64IZBB-NEXT:    addi a0, a0, -273
; RV64IZBB-NEXT:    slli a0, a0, 14
; RV64IZBB-NEXT:    addi a0, a0, -1093
; RV64IZBB-NEXT:    ret
;
; RV64IZBS-LABEL: imm_neg_9223301666034697285:
; RV64IZBS:         # %bb.0:
; RV64IZBS-NEXT:    lui a0, 506812
; RV64IZBS-NEXT:    addiw a0, a0, -1093
; RV64IZBS-NEXT:    bseti a0, a0, 46
; RV64IZBS-NEXT:    bseti a0, a0, 63
; RV64IZBS-NEXT:    ret
;
; RV64IXTHEADBB-LABEL: imm_neg_9223301666034697285:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 917505
; RV64IXTHEADBB-NEXT:  slli a0, a0, 8
; RV64IXTHEADBB-NEXT:  addi a0, a0, 31
; RV64IXTHEADBB-NEXT:  slli a0, a0, 12
; RV64IXTHEADBB-NEXT:  addi a0, a0, -273
; RV64IXTHEADBB-NEXT:  slli a0, a0, 14
; RV64IXTHEADBB-NEXT:  addi a0, a0, -1093

```

```
; RV64IXTHEADBB-NEXT:  ret
ret i64 -9223301666034697285 ; 0x800040007bbbbbbb
}
```

```
define i64 @imm_neg_2219066437() {
; RV64I-LABEL: imm_neg_2219066437:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a0, 913135
; RV64I-NEXT:  slli a0, a0, 2
; RV64I-NEXT:  addi a0, a0, -1093
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_neg_2219066437:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 913135
; RV64IZBA-NEXT:  slli a0, a0, 2
; RV64IZBA-NEXT:  addi a0, a0, -1093
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_neg_2219066437:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 913135
; RV64IZBB-NEXT:  slli a0, a0, 2
; RV64IZBB-NEXT:  addi a0, a0, -1093
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_neg_2219066437:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 913135
; RV64IZBS-NEXT:  slli a0, a0, 2
; RV64IZBS-NEXT:  addi a0, a0, -1093
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_neg_2219066437:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 913135
; RV64IXTHEADBB-NEXT:
    slli a0, a0, 2
; RV64IXTHEADBB-NEXT:  addi a0, a0, -1093
; RV64IXTHEADBB-NEXT:  ret
ret i64 -2219066437 ; 0xffffffff7bbbbbbb
}
```

```
define i64 @imm_neg_8798043653189() {
; RV64I-LABEL: imm_neg_8798043653189:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a0, 917475
; RV64I-NEXT:  addiw a0, a0, -273
```

```

; RV64I-NEXT:  slli a0, a0, 14
; RV64I-NEXT:  addi a0, a0, -1093
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_neg_8798043653189:
; RV64IZBA:      # %bb.0:
; RV64IZBA-NEXT:  lui a0, 917475
; RV64IZBA-NEXT:  addiw a0, a0, -273
; RV64IZBA-NEXT:  slli a0, a0, 14
; RV64IZBA-NEXT:  addi a0, a0, -1093
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_neg_8798043653189:
; RV64IZBB:      # %bb.0:
; RV64IZBB-NEXT:  lui a0, 917475
; RV64IZBB-NEXT:  addiw a0, a0, -273
; RV64IZBB-NEXT:  slli a0, a0, 14
; RV64IZBB-NEXT:  addi a0, a0, -1093
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_neg_8798043653189:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT:  lui a0, 572348
; RV64IZBS-NEXT:
;   addiw a0, a0, -1093
; RV64IZBS-NEXT:  bclri a0, a0, 43
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_neg_8798043653189:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 917475
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -273
; RV64IXTHEADBB-NEXT:  slli a0, a0, 14
; RV64IXTHEADBB-NEXT:  addi a0, a0, -1093
; RV64IXTHEADBB-NEXT:  ret
;   ret i64 -8798043653189 ; 0xffff7ff8bbbbbbb
}

```

```

define i64 @imm_9223372034904144827() {
; RV64I-LABEL: imm_9223372034904144827:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a0, 572348
; RV64I-NEXT:  addiw a0, a0, -1093
; RV64I-NEXT:  slli a1, a0, 63
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_9223372034904144827:

```

```

; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 572348
; RV64IZBA-NEXT:  addiw a0, a0, -1093
; RV64IZBA-NEXT:  slli a1, a0, 63
; RV64IZBA-NEXT:  add a0, a0, a1
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_9223372034904144827:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:
    lui a0, 572348
; RV64IZBB-NEXT:  addiw a0, a0, -1093
; RV64IZBB-NEXT:  slli a1, a0, 63
; RV64IZBB-NEXT:  add a0, a0, a1
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_9223372034904144827:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 572348
; RV64IZBS-NEXT:  addiw a0, a0, -1093
; RV64IZBS-NEXT:  bclri a0, a0, 63
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_9223372034904144827:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 572348
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -1093
; RV64IXTHEADBB-NEXT:  slli a1, a0, 63
; RV64IXTHEADBB-NEXT:  add a0, a0, a1
; RV64IXTHEADBB-NEXT:  ret
    ret i64 9223372034904144827 ; 0x7fffffff8bbbbbbb
}

```

```

define i64 @imm_neg_9223354442718100411() {
; RV64-NOPOOL-LABEL: imm_neg_9223354442718100411:
; RV64-NOPOOL:    # %bb.0:
; RV64-NOPOOL-NEXT:  lui a0, 524287
; RV64-NOPOOL-NEXT:  slli a0, a0, 6
; RV64-NOPOOL-NEXT:  addi a0, a0, -29
; RV64-NOPOOL-NEXT:  slli a0, a0, 12
; RV64-NOPOOL-NEXT:
    addi a0, a0, -273
; RV64-NOPOOL-NEXT:  slli a0, a0, 14
; RV64-NOPOOL-NEXT:  addi a0, a0, -1093
; RV64-NOPOOL-NEXT:  ret
;
; RV64I-POOL-LABEL: imm_neg_9223354442718100411:
; RV64I-POOL:    # %bb.0:

```



```

; RV64I-POOL-NEXT: lui a0, %hi(.LCPI43_0)
; RV64I-POOL-NEXT: ld a0, %lo(.LCPI43_0)(a0)
; RV64I-POOL-NEXT: ret
;
; RV64IZBA-LABEL: imm_neg_9223354442718100411:
; RV64IZBA: # %bb.0:
; RV64IZBA-NEXT: lui a0, 524287
; RV64IZBA-NEXT: slli a0, a0, 6
; RV64IZBA-NEXT: addi a0, a0, -29
; RV64IZBA-NEXT: slli a0, a0, 12
; RV64IZBA-NEXT: addi a0, a0, -273
; RV64IZBA-NEXT: slli a0, a0, 14
; RV64IZBA-NEXT: addi a0, a0, -1093
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: imm_neg_9223354442718100411:
; RV64IZBB: # %bb.0:
; RV64IZBB-NEXT: lui a0, 524287
; RV64IZBB-NEXT: slli a0, a0, 6
; RV64IZBB-NEXT: addi a0, a0, -29
; RV64IZBB-NEXT: slli a0, a0, 12
; RV64IZBB-NEXT: addi a0, a0, -273
; RV64IZBB-NEXT: slli a0, a0, 14
; RV64IZBB-NEXT:
; addi a0, a0, -1093
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: imm_neg_9223354442718100411:
; RV64IZBS: # %bb.0:
; RV64IZBS-NEXT: lui a0, 572348
; RV64IZBS-NEXT: addiw a0, a0, -1093
; RV64IZBS-NEXT: bclri a0, a0, 44
; RV64IZBS-NEXT: bclri a0, a0, 63
; RV64IZBS-NEXT: ret
;
; RV64IXTHEADBB-LABEL: imm_neg_9223354442718100411:
; RV64IXTHEADBB: # %bb.0:
; RV64IXTHEADBB-NEXT: lui a0, 524287
; RV64IXTHEADBB-NEXT: slli a0, a0, 6
; RV64IXTHEADBB-NEXT: addi a0, a0, -29
; RV64IXTHEADBB-NEXT: slli a0, a0, 12
; RV64IXTHEADBB-NEXT: addi a0, a0, -273
; RV64IXTHEADBB-NEXT: slli a0, a0, 14
; RV64IXTHEADBB-NEXT: addi a0, a0, -1093
; RV64IXTHEADBB-NEXT: ret
ret i64 9223354442718100411 ; 0x7fffefff8bbbbbbb
}

```

```

define i64 @imm_2863311530() {
; RV64I-LABEL: imm_2863311530:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 349525
; RV64I-NEXT:  addiw a0, a0, 1365
; RV64I-NEXT:  slli a0, a0, 1
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_2863311530:
; RV64IZBA:
;   # %bb.0:
; RV64IZBA-NEXT:  lui a0, 349525
; RV64IZBA-NEXT:  addiw a0, a0, 1365
; RV64IZBA-NEXT:  slli a0, a0, 1
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_2863311530:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 349525
; RV64IZBB-NEXT:  addiw a0, a0, 1365
; RV64IZBB-NEXT:  slli a0, a0, 1
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_2863311530:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 349525
; RV64IZBS-NEXT:  addiw a0, a0, 1365
; RV64IZBS-NEXT:  slli a0, a0, 1
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_2863311530:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 349525
; RV64IXTHEADBB-NEXT:  addiw a0, a0, 1365
; RV64IXTHEADBB-NEXT:  slli a0, a0, 1
; RV64IXTHEADBB-NEXT:  ret
ret i64 2863311530 ; #0xaaaaaaaa
}

```

```

define i64 @imm_neg_2863311530() {
; RV64I-LABEL: imm_neg_2863311530:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 699051
; RV64I-NEXT:  addiw a0, a0, -1365
; RV64I-NEXT:  slli a0,
a0, 1
; RV64I-NEXT:  ret
;

```

```

; RV64IZBA-LABEL: imm_neg_2863311530:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 699051
; RV64IZBA-NEXT:  addiw a0, a0, -1365
; RV64IZBA-NEXT:  slli a0, a0, 1
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_neg_2863311530:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 699051
; RV64IZBB-NEXT:  addiw a0, a0, -1365
; RV64IZBB-NEXT:  slli a0, a0, 1
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_neg_2863311530:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 699051
; RV64IZBS-NEXT:  addiw a0, a0, -1365
; RV64IZBS-NEXT:  slli a0, a0, 1
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_neg_2863311530:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 699051
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -1365
; RV64IXTHEADBB-NEXT:  slli a0, a0, 1
; RV64IXTHEADBB-NEXT:  ret
ret i64 -2863311530 ; #0xffffffff55555556
}

```

```

define i64 @imm_2147486378() {
; RV64I-LABEL: imm_2147486378:
; RV64I:    # %bb.0:
; RV64I-NEXT:
    li a0, 1
; RV64I-NEXT:  slli a0, a0, 31
; RV64I-NEXT:  addi a0, a0, 1365
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_2147486378:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  li a0, 1
; RV64IZBA-NEXT:  slli a0, a0, 31
; RV64IZBA-NEXT:  addi a0, a0, 1365
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_2147486378:
; RV64IZBB:    # %bb.0:

```

```

; RV64IZBB-NEXT:  li a0, 1
; RV64IZBB-NEXT:  slli a0, a0, 31
; RV64IZBB-NEXT:  addi a0, a0, 1365
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_2147486378:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT: li a0, 1365
; RV64IZBS-NEXT: bseti a0, a0, 31
; RV64IZBS-NEXT: ret
;
; RV64IXTHEADBB-LABEL: imm_2147486378:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT: li a0, 1
; RV64IXTHEADBB-NEXT: slli a0, a0, 31
; RV64IXTHEADBB-NEXT: addi a0, a0, 1365
; RV64IXTHEADBB-NEXT: ret
ret i64 2147485013
}

define i64 @imm_neg_2147485013() {
; RV64I-LABEL: imm_neg_2147485013:
; RV64I:      # %bb.0:
; RV64I-NEXT:
    lui a0, 524288
; RV64I-NEXT:  addi a0, a0, -1365
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_neg_2147485013:
; RV64IZBA:      # %bb.0:
; RV64IZBA-NEXT: lui a0, 524288
; RV64IZBA-NEXT: addi a0, a0, -1365
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: imm_neg_2147485013:
; RV64IZBB:      # %bb.0:
; RV64IZBB-NEXT: lui a0, 524288
; RV64IZBB-NEXT: addi a0, a0, -1365
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: imm_neg_2147485013:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT: lui a0, 524288
; RV64IZBS-NEXT: addi a0, a0, -1365
; RV64IZBS-NEXT: ret
;
; RV64IXTHEADBB-LABEL: imm_neg_2147485013:
; RV64IXTHEADBB:  # %bb.0:

```

```

; RV64IXTHEADBB-NEXT: lui a0, 524288
; RV64IXTHEADBB-NEXT: addi a0, a0, -1365
; RV64IXTHEADBB-NEXT: ret
ret i64 -2147485013
}

define i64 @imm_12900924131259() {
; RV64I-LABEL: imm_12900924131259:
; RV64I:    # %bb.0:
; RV64I-NEXT: lui a0, 188
; RV64I-NEXT: addiw a0, a0, -1093
; RV64I-NEXT: slli a0, a0, 24
; RV64I-NEXT: addi a0,
a0, 1979
; RV64I-NEXT: ret
;
; RV64IZBA-LABEL: imm_12900924131259:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT: lui a0, 768955
; RV64IZBA-NEXT: slli.uw a0, a0, 12
; RV64IZBA-NEXT: addi a0, a0, 1979
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: imm_12900924131259:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT: lui a0, 188
; RV64IZBB-NEXT: addiw a0, a0, -1093
; RV64IZBB-NEXT: slli a0, a0, 24
; RV64IZBB-NEXT: addi a0, a0, 1979
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: imm_12900924131259:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT: lui a0, 188
; RV64IZBS-NEXT: addiw a0, a0, -1093
; RV64IZBS-NEXT: slli a0, a0, 24
; RV64IZBS-NEXT: addi a0, a0, 1979
; RV64IZBS-NEXT: ret
;
; RV64IXTHEADBB-LABEL: imm_12900924131259:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT: lui a0, 188
; RV64IXTHEADBB-NEXT: addiw a0, a0, -1093
; RV64IXTHEADBB-NEXT: slli a0, a0, 24
; RV64IXTHEADBB-NEXT: addi a0, a0, 1979
; RV64IXTHEADBB-NEXT: ret
ret i64 12900924131259
}

```

```

define
i64 @imm_50394234880() {
; RV64I-LABEL: imm_50394234880:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 188
; RV64I-NEXT:  addiw a0, a0, -1093
; RV64I-NEXT:  slli a0, a0, 16
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_50394234880:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 768955
; RV64IZBA-NEXT:  slli.uw a0, a0, 4
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_50394234880:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 188
; RV64IZBB-NEXT:  addiw a0, a0, -1093
; RV64IZBB-NEXT:  slli a0, a0, 16
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_50394234880:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 188
; RV64IZBS-NEXT:  addiw a0, a0, -1093
; RV64IZBS-NEXT:  slli a0, a0, 16
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_50394234880:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 188
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -1093
; RV64IXTHEADBB-NEXT:  slli a0, a0, 16
; RV64IXTHEADBB-NEXT:  ret
ret
i64 50394234880
}

define i64 @imm_12900936431479() {
; RV64I-LABEL: imm_12900936431479:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 192239
; RV64I-NEXT:  slli a0, a0, 2
; RV64I-NEXT:  addi a0, a0, -1093
; RV64I-NEXT:  slli a0, a0, 12
; RV64I-NEXT:  addi a0, a0, 1911

```

```

; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_12900936431479:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 768956
; RV64IZBA-NEXT:  addi a0, a0, -1093
; RV64IZBA-NEXT:  slli.uw a0, a0, 12
; RV64IZBA-NEXT:  addi a0, a0, 1911
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_12900936431479:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 192239
; RV64IZBB-NEXT:  slli a0, a0, 2
; RV64IZBB-NEXT:  addi a0, a0, -1093
; RV64IZBB-NEXT:  slli a0, a0, 12
; RV64IZBB-NEXT:  addi a0, a0, 1911
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_12900936431479:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 192239
; RV64IZBS-NEXT:  slli a0, a0, 2
; RV64IZBS-NEXT:  addi a0, a0, -1093
; RV64IZBS-NEXT:
;               slli a0, a0, 12
; RV64IZBS-NEXT:  addi a0, a0, 1911
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_12900936431479:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 192239
; RV64IXTHEADBB-NEXT:  slli a0, a0, 2
; RV64IXTHEADBB-NEXT:  addi a0, a0, -1093
; RV64IXTHEADBB-NEXT:  slli a0, a0, 12
; RV64IXTHEADBB-NEXT:  addi a0, a0, 1911
; RV64IXTHEADBB-NEXT:  ret
ret i64 12900936431479
}

```

```

define i64 @imm_12900918536874() {
; RV64I-LABEL: imm_12900918536874:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 384477
; RV64I-NEXT:  addiw a0, a0, 1365
; RV64I-NEXT:  slli a0, a0, 12
; RV64I-NEXT:  addi a0, a0, 1365
; RV64I-NEXT:  slli a0, a0, 1

```

```

; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_12900918536874:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 768955
; RV64IZBA-NEXT:  addi a0, a0, -1365
; RV64IZBA-NEXT:  slli.uw a0, a0, 12
; RV64IZBA-NEXT:  addi a0, a0, -1366
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_12900918536874:
;
RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 384477
; RV64IZBB-NEXT:  addiw a0, a0, 1365
; RV64IZBB-NEXT:  slli a0, a0, 12
; RV64IZBB-NEXT:  addi a0, a0, 1365
; RV64IZBB-NEXT:  slli a0, a0, 1
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_12900918536874:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 384477
; RV64IZBS-NEXT:  addiw a0, a0, 1365
; RV64IZBS-NEXT:  slli a0, a0, 12
; RV64IZBS-NEXT:  addi a0, a0, 1365
; RV64IZBS-NEXT:  slli a0, a0, 1
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_12900918536874:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 384477
; RV64IXTHEADBB-NEXT:  addiw a0, a0, 1365
; RV64IXTHEADBB-NEXT:  slli a0, a0, 12
; RV64IXTHEADBB-NEXT:  addi a0, a0, 1365
; RV64IXTHEADBB-NEXT:  slli a0, a0, 1
; RV64IXTHEADBB-NEXT:  ret
ret i64 12900918536874
}

define i64 @imm_12900925247761() {
; RV64I-LABEL: imm_12900925247761:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 384478
; RV64I-NEXT:  addiw
a0, a0, -1911
; RV64I-NEXT:  slli a0, a0, 13
; RV64I-NEXT:  addi a0, a0, -2048

```



```

; RV64I-NEXT:  addi a0, a0, -1775
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_12900925247761:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 768955
; RV64IZBA-NEXT:  addi a0, a0, 273
; RV64IZBA-NEXT:  slli.uw a0, a0, 12
; RV64IZBA-NEXT:  addi a0, a0, 273
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_12900925247761:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 384478
; RV64IZBB-NEXT:  addiw a0, a0, -1911
; RV64IZBB-NEXT:  slli a0, a0, 13
; RV64IZBB-NEXT:  addi a0, a0, -2048
; RV64IZBB-NEXT:  addi a0, a0, -1775
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_12900925247761:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 384478
; RV64IZBS-NEXT:  addiw a0, a0, -1911
; RV64IZBS-NEXT:  slli a0, a0, 13
; RV64IZBS-NEXT:  addi a0, a0, -2048
; RV64IZBS-NEXT:  addi a0, a0, -1775
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_12900925247761:
; RV64IXTHEADBB:
;   # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 384478
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -1911
; RV64IXTHEADBB-NEXT:  slli a0, a0, 13
; RV64IXTHEADBB-NEXT:  addi a0, a0, -2048
; RV64IXTHEADBB-NEXT:  addi a0, a0, -1775
; RV64IXTHEADBB-NEXT:  ret
ret i64 12900925247761
}

```

```

define i64 @imm_7158272001() {
; RV64I-LABEL: imm_7158272001:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 427
; RV64I-NEXT:  addiw a0, a0, -1367
; RV64I-NEXT:  slli a0, a0, 12
; RV64I-NEXT:  addi a0, a0, 1

```

```

; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_7158272001:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 349525
; RV64IZBA-NEXT:  sh2add a0, a0, a0
; RV64IZBA-NEXT:  addi a0, a0, 1
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_7158272001:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 427
; RV64IZBB-NEXT:  addiw a0, a0, -1367
; RV64IZBB-NEXT:  slli a0, a0, 12
; RV64IZBB-NEXT:  addi a0, a0, 1
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_7158272001:
; RV64IZBS:
;   # %bb.0:
; RV64IZBS-NEXT:  lui a0, 427
; RV64IZBS-NEXT:  addiw a0, a0, -1367
; RV64IZBS-NEXT:  slli a0, a0, 12
; RV64IZBS-NEXT:  addi a0, a0, 1
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_7158272001:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 427
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -1367
; RV64IXTHEADBB-NEXT:  slli a0, a0, 12
; RV64IXTHEADBB-NEXT:  addi a0, a0, 1
; RV64IXTHEADBB-NEXT:  ret
ret i64 7158272001 ; 0x0000_0001_aaaa_9001
}

```

```

define i64 @imm_12884889601() {
; RV64I-LABEL: imm_12884889601:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a0, 768
; RV64I-NEXT:  addiw a0, a0, -3
; RV64I-NEXT:  slli a0, a0, 12
; RV64I-NEXT:  addi a0, a0, 1
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_12884889601:
; RV64IZBA:  # %bb.0:
; RV64IZBA-NEXT:  lui a0, 349525

```

```

; RV64IZBA-NEXT: sh3add a0, a0, a0
; RV64IZBA-NEXT: addi a0, a0, 1
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: imm_12884889601:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:
    lui a0, 768
; RV64IZBB-NEXT: addiw a0, a0, -3
; RV64IZBB-NEXT: slli a0, a0, 12
; RV64IZBB-NEXT: addi a0, a0, 1
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: imm_12884889601:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT: lui a0, 768
; RV64IZBS-NEXT: addiw a0, a0, -3
; RV64IZBS-NEXT: slli a0, a0, 12
; RV64IZBS-NEXT: addi a0, a0, 1
; RV64IZBS-NEXT: ret
;
; RV64IXTHEADBB-LABEL: imm_12884889601:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT: lui a0, 768
; RV64IXTHEADBB-NEXT: addiw a0, a0, -3
; RV64IXTHEADBB-NEXT: slli a0, a0, 12
; RV64IXTHEADBB-NEXT: addi a0, a0, 1
; RV64IXTHEADBB-NEXT: ret
ret i64 12884889601 ; 0x0000_0002_ffff_d001
}

```

```

define i64 @imm_neg_3435982847() {
; RV64I-LABEL: imm_neg_3435982847:
; RV64I:    # %bb.0:
; RV64I-NEXT: lui a0, 1048371
; RV64I-NEXT: addiw a0, a0, 817
; RV64I-NEXT: slli a0, a0, 12
; RV64I-NEXT: addi a0, a0, 1
; RV64I-NEXT: ret
;
; RV64IZBA-LABEL: imm_neg_3435982847:
; RV64IZBA:
    # %bb.0:
; RV64IZBA-NEXT: lui a0, 768955
; RV64IZBA-NEXT: sh1add a0, a0, a0
; RV64IZBA-NEXT: addi a0, a0, 1
; RV64IZBA-NEXT: ret
;

```

```

; RV64IZBB-LABEL: imm_neg_3435982847:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 1048371
; RV64IZBB-NEXT:  addiw a0, a0, 817
; RV64IZBB-NEXT:  slli a0, a0, 12
; RV64IZBB-NEXT:  addi a0, a0, 1
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_neg_3435982847:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  lui a0, 734001
; RV64IZBS-NEXT:  addiw a0, a0, 1
; RV64IZBS-NEXT:  bclri a0, a0, 31
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_neg_3435982847:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 1048371
; RV64IXTHEADBB-NEXT:  addiw a0, a0, 817
; RV64IXTHEADBB-NEXT:  slli a0, a0, 12
; RV64IXTHEADBB-NEXT:  addi a0, a0, 1
; RV64IXTHEADBB-NEXT:  ret
ret i64 -3435982847 ; 0xffff_ffff_3333_1001
}

```

```

define i64 @imm_neg_5726842879() {
; RV64I-LABEL: imm_neg_5726842879:
; RV64I:    # %bb.0:
; RV64I-NEXT:
    lui a0, 1048235
; RV64I-NEXT:  addiw a0, a0, -1419
; RV64I-NEXT:  slli a0, a0, 12
; RV64I-NEXT:  addi a0, a0, 1
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_neg_5726842879:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  lui a0, 768945
; RV64IZBA-NEXT:  sh2add a0, a0, a0
; RV64IZBA-NEXT:  addi a0, a0, 1
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_neg_5726842879:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  lui a0, 1048235
; RV64IZBB-NEXT:  addiw a0, a0, -1419
; RV64IZBB-NEXT:  slli a0, a0, 12
; RV64IZBB-NEXT:  addi a0, a0, 1

```

```

; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_neg_5726842879:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT:  lui a0, 698997
; RV64IZBS-NEXT:  addiw a0, a0, 1
; RV64IZBS-NEXT:  bclri a0, a0, 32
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm_neg_5726842879:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 1048235
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -1419
; RV64IXTHEADBB-NEXT:  slli a0, a0, 12
; RV64IXTHEADBB-NEXT:
    addi a0, a0, 1
; RV64IXTHEADBB-NEXT:  ret
ret i64 -5726842879 ; 0xffff_fffe_aaa7_5001
}

```

```

define i64 @imm_neg_10307948543() {
; RV64I-LABEL: imm_neg_10307948543:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a0, 1047962
; RV64I-NEXT:  addiw a0, a0, -1645
; RV64I-NEXT:  slli a0, a0, 12
; RV64I-NEXT:  addi a0, a0, 1
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: imm_neg_10307948543:
; RV64IZBA:      # %bb.0:
; RV64IZBA-NEXT:  lui a0, 768955
; RV64IZBA-NEXT:  sh3add a0, a0, a0
; RV64IZBA-NEXT:  addi a0, a0, 1
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm_neg_10307948543:
; RV64IZBB:      # %bb.0:
; RV64IZBB-NEXT:  lui a0, 1047962
; RV64IZBB-NEXT:  addiw a0, a0, -1645
; RV64IZBB-NEXT:  slli a0, a0, 12
; RV64IZBB-NEXT:  addi a0, a0, 1
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm_neg_10307948543:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT:  lui a0, 629139
; RV64IZBS-NEXT:  addiw a0, a0, 1

```

```

; RV64IZBS-NEXT:  bclri a0, a0, 33
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL:
imm_neg_10307948543:
; RV64IXTHEADBB:   # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 1047962
; RV64IXTHEADBB-NEXT:  addiw a0, a0, -1645
; RV64IXTHEADBB-NEXT:  slli a0, a0, 12
; RV64IXTHEADBB-NEXT:  addi a0, a0, 1
; RV64IXTHEADBB-NEXT:  ret
ret i64 -10307948543 ; 0xffff_fffd_9999_3001
}

```

```

define i64 @li_ropi_1() {
; RV64I-LABEL: li_ropi_1:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a0, -17
; RV64I-NEXT:  slli a0, a0, 43
; RV64I-NEXT:  addi a0, a0, -1
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: li_ropi_1:
; RV64IZBA:   # %bb.0:
; RV64IZBA-NEXT:  li a0, -17
; RV64IZBA-NEXT:  slli a0, a0, 43
; RV64IZBA-NEXT:  addi a0, a0, -1
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: li_ropi_1:
; RV64IZBB:   # %bb.0:
; RV64IZBB-NEXT:  li a0, -18
; RV64IZBB-NEXT:  rori a0, a0, 21
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: li_ropi_1:
; RV64IZBS:   # %bb.0:
; RV64IZBS-NEXT:  li a0, -17
; RV64IZBS-NEXT:  slli a0, a0, 43
; RV64IZBS-NEXT:  addi a0, a0,
-1
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: li_ropi_1:
; RV64IXTHEADBB:   # %bb.0:
; RV64IXTHEADBB-NEXT:  li a0, -18
; RV64IXTHEADBB-NEXT:  th.srri a0, a0, 21
; RV64IXTHEADBB-NEXT:  ret

```

```

ret i64 -149533581377537
}

define i64 @li_ropi_2() {
; RV64I-LABEL: li_ropi_2:
; RV64I:      # %bb.0:
; RV64I-NEXT: li a0, -5
; RV64I-NEXT: slli a0, a0, 60
; RV64I-NEXT: addi a0, a0, -6
; RV64I-NEXT: ret
;
; RV64IZBA-LABEL: li_ropi_2:
; RV64IZBA:   # %bb.0:
; RV64IZBA-NEXT: li a0, -5
; RV64IZBA-NEXT: slli a0, a0, 60
; RV64IZBA-NEXT: addi a0, a0, -6
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: li_ropi_2:
; RV64IZBB:   # %bb.0:
; RV64IZBB-NEXT: li a0, -86
; RV64IZBB-NEXT: rori a0, a0, 4
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: li_ropi_2:
; RV64IZBS:   # %bb.0:
; RV64IZBS-NEXT: li a0, -5
; RV64IZBS-NEXT: slli a0, a0, 60
; RV64IZBS-NEXT: addi a0, a0, -6
; RV64IZBS-NEXT: ret
;
; RV64IXTHEADBB-LABEL: li_ropi_2:
; RV64IXTHEADBB:
;   # %bb.0:
; RV64IXTHEADBB-NEXT: li a0, -86
; RV64IXTHEADBB-NEXT: th.srri a0, a0, 4
; RV64IXTHEADBB-NEXT: ret
ret i64 -5764607523034234886
}

```

```

define i64 @li_ropi_3() {
; RV64I-LABEL: li_ropi_3:
; RV64I:      # %bb.0:
; RV64I-NEXT: li a0, -17
; RV64I-NEXT: slli a0, a0, 27
; RV64I-NEXT: addi a0, a0, -1
; RV64I-NEXT: ret
;

```

```

; RV64IZBA-LABEL: li_ropi_3:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT: li a0, -17
; RV64IZBA-NEXT: slli a0, a0, 27
; RV64IZBA-NEXT: addi a0, a0, -1
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: li_ropi_3:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT: li a0, -18
; RV64IZBB-NEXT: rori a0, a0, 37
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: li_ropi_3:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT: li a0, -17
; RV64IZBS-NEXT: slli a0, a0, 27
; RV64IZBS-NEXT: addi a0, a0, -1
; RV64IZBS-NEXT: ret
;
; RV64IXTHEADBB-LABEL: li_ropi_3:
; RV64IXTHEADBB:    # %bb.0:
; RV64IXTHEADBB-NEXT: li a0, -18
; RV64IXTHEADBB-NEXT: th.srri a0,
a0, 37
; RV64IXTHEADBB-NEXT: ret
ret i64 -2281701377
}

```

; This used to assert when compiled with Zba.

```

define i64 @PR54812() {
; RV64I-LABEL: PR54812:
; RV64I:    # %bb.0:
; RV64I-NEXT: lui a0, 1048447
; RV64I-NEXT: addiw a0, a0, 1407
; RV64I-NEXT: slli a0, a0, 12
; RV64I-NEXT: ret
;
; RV64IZBA-LABEL: PR54812:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT: lui a0, 872917
; RV64IZBA-NEXT: sh1add a0, a0, a0
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: PR54812:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT: lui a0, 1048447
; RV64IZBB-NEXT: addiw a0, a0, 1407

```



```

; RV64IZBB-NEXT:  slli a0, a0, 12
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: PR54812:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT:  lui a0, 1045887
; RV64IZBS-NEXT:  bclri a0, a0, 31
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: PR54812:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 1048447
; RV64IXTHEADBB-NEXT:  addiw a0, a0, 1407
; RV64IXTHEADBB-NEXT:  slli a0, a0, 12
; RV64IXTHEADBB-NEXT:
    ret
ret i64 -2158497792;
}

```

```

define signext i32 @pos_2048() nounwind {

```

```

; RV64I-LABEL: pos_2048:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a0, 1
; RV64I-NEXT:  slli a0, a0, 11
; RV64I-NEXT:  ret
;
; RV64IZBA-LABEL: pos_2048:
; RV64IZBA:    # %bb.0:
; RV64IZBA-NEXT:  li a0, 1
; RV64IZBA-NEXT:  slli a0, a0, 11
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: pos_2048:
; RV64IZBB:    # %bb.0:
; RV64IZBB-NEXT:  li a0, 1
; RV64IZBB-NEXT:  slli a0, a0, 11
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: pos_2048:
; RV64IZBS:    # %bb.0:
; RV64IZBS-NEXT:  bseti a0, zero, 11
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: pos_2048:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  li a0, 1
; RV64IXTHEADBB-NEXT:  slli a0, a0, 11
; RV64IXTHEADBB-NEXT:  ret

```

```

ret i32 2048
}

define i64 @imm64_same_lo_hi() nounwind {
; RV64I-LABEL: imm64_same_lo_hi:
; RV64I:      # %bb.0:
; RV64I-NEXT: lui a0, 65793
; RV64I-NEXT: addiw a0, a0, 16
; RV64I-NEXT:
; slli a1, a0, 32
; RV64I-NEXT: add a0, a0, a1
; RV64I-NEXT: ret
;
; RV64IZBA-LABEL: imm64_same_lo_hi:
; RV64IZBA:   # %bb.0:
; RV64IZBA-NEXT: lui a0, 65793
; RV64IZBA-NEXT: addiw a0, a0, 16
; RV64IZBA-NEXT: slli a1, a0, 32
; RV64IZBA-NEXT: add a0, a0, a1
; RV64IZBA-NEXT: ret
;
; RV64IZBB-LABEL: imm64_same_lo_hi:
; RV64IZBB:   # %bb.0:
; RV64IZBB-NEXT: lui a0, 65793
; RV64IZBB-NEXT: addiw a0, a0, 16
; RV64IZBB-NEXT: slli a1, a0, 32
; RV64IZBB-NEXT: add a0, a0, a1
; RV64IZBB-NEXT: ret
;
; RV64IZBS-LABEL: imm64_same_lo_hi:
; RV64IZBS:   # %bb.0:
; RV64IZBS-NEXT: lui a0, 65793
; RV64IZBS-NEXT: addiw a0, a0, 16
; RV64IZBS-NEXT: slli a1, a0, 32
; RV64IZBS-NEXT: add a0, a0, a1
; RV64IZBS-NEXT: ret
;
; RV64IXTHEADBB-LABEL: imm64_same_lo_hi:
; RV64IXTHEADBB:   # %bb.0:
; RV64IXTHEADBB-NEXT: lui a0, 65793
; RV64IXTHEADBB-NEXT: addiw a0, a0, 16
; RV64IXTHEADBB-NEXT: slli a1, a0, 32
; RV64IXTHEADBB-NEXT: add a0, a0, a1
;
; RV64IXTHEADBB-NEXT: ret
ret i64 1157442765409226768 ; 0x0101010101010101
}

```

```

; Same as above with optsize. Make sure we use constant pool on RV64
define i64 @imm64_same_lo_hi_optsize() nounwind optsize {
; RV64-NOPOOL-LABEL: imm64_same_lo_hi_optsize:
; RV64-NOPOOL:      # %bb.0:
; RV64-NOPOOL-NEXT:  lui a0, 65793
; RV64-NOPOOL-NEXT:  addiw a0, a0, 16
; RV64-NOPOOL-NEXT:  slli a1, a0, 32
; RV64-NOPOOL-NEXT:  add a0, a0, a1
; RV64-NOPOOL-NEXT:  ret
;
; RV64I-POOL-LABEL: imm64_same_lo_hi_optsize:
; RV64I-POOL:      # %bb.0:
; RV64I-POOL-NEXT:  lui a0, %hi(.LCPI64_0)
; RV64I-POOL-NEXT:  ld a0, %lo(.LCPI64_0)(a0)
; RV64I-POOL-NEXT:  ret
;
; RV64IZBA-LABEL: imm64_same_lo_hi_optsize:
; RV64IZBA:      # %bb.0:
; RV64IZBA-NEXT:  lui a0, 65793
; RV64IZBA-NEXT:  addiw a0, a0, 16
; RV64IZBA-NEXT:  slli a1, a0, 32
; RV64IZBA-NEXT:  add a0, a0, a1
; RV64IZBA-NEXT:  ret
;
; RV64IZBB-LABEL: imm64_same_lo_hi_optsize:
; RV64IZBB:      # %bb.0:
; RV64IZBB-NEXT:  lui
a0, 65793
; RV64IZBB-NEXT:  addiw a0, a0, 16
; RV64IZBB-NEXT:  slli a1, a0, 32
; RV64IZBB-NEXT:  add a0, a0, a1
; RV64IZBB-NEXT:  ret
;
; RV64IZBS-LABEL: imm64_same_lo_hi_optsize:
; RV64IZBS:      # %bb.0:
; RV64IZBS-NEXT:  lui a0, 65793
; RV64IZBS-NEXT:  addiw a0, a0, 16
; RV64IZBS-NEXT:  slli a1, a0, 32
; RV64IZBS-NEXT:  add a0, a0, a1
; RV64IZBS-NEXT:  ret
;
; RV64IXTHEADBB-LABEL: imm64_same_lo_hi_optsize:
; RV64IXTHEADBB:  # %bb.0:
; RV64IXTHEADBB-NEXT:  lui a0, 65793
; RV64IXTHEADBB-NEXT:  addiw a0, a0, 16
; RV64IXTHEADBB-NEXT:  slli a1, a0, 32
; RV64IXTHEADBB-NEXT:  add a0, a0, a1
; RV64IXTHEADBB-NEXT:  ret

```

```
ret i64 1157442765409226768 ; 0x0101010101010101
}
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck -check-prefix=RV64I %s
```

; Check indexed and unindexed, sext, zext and anyext loads

```
define void @lb(ptr %a, ptr %b) nounwind {
; RV64I-LABEL: lb:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lb a2, 1(a0)
; RV64I-NEXT:  lbu zero, 0(a0)
; RV64I-NEXT:  sw a2, 0(a1)
; RV64I-NEXT:  ret
%1 = getelementptr i8, ptr %a, i32 1
%2 = load i8, ptr %1
%3 = sext i8 %2 to i32
; the unused load will produce an anyext for selection
%4 = load volatile i8, ptr %a
store i32 %3, ptr %b
ret void
}
```

```
define void @lbu(ptr %a, ptr %b) nounwind {
; RV64I-LABEL: lbu:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lbu a0, 1(a0)
; RV64I-NEXT:  sw a0, 0(a1)
; RV64I-NEXT:  ret
%1 = getelementptr i8, ptr %a, i32 1
%2 = load i8, ptr %1
%3 = zext i8 %2 to i32
store i32 %3, ptr %b
ret void
}
```

```
define void @lh(ptr
%a, ptr %b) nounwind {
; RV64I-LABEL: lh:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lh a2, 2(a0)
; RV64I-NEXT:  lh zero, 0(a0)
; RV64I-NEXT:  sw a2, 0(a1)
; RV64I-NEXT:  ret
%1 = getelementptr i16, ptr %a, i32 1
%2 = load i16, ptr %1
%3 = sext i16 %2 to i32
```

```

; the unused load will produce an anyext for selection
%4 = load volatile i16, ptr %a
store i32 %3, ptr %b
ret void
}

```

```

define void @lhu(ptr %a, ptr %b) nounwind {
; RV64I-LABEL: lhu:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lhu a0, 2(a0)
; RV64I-NEXT:  sw a0, 0(a1)
; RV64I-NEXT:  ret
%1 = getelementptr i16, ptr %a, i32 1
%2 = load i16, ptr %1
%3 = zext i16 %2 to i32
store i32 %3, ptr %b
ret void
}

```

```

define void @lw(ptr %a, ptr %b) nounwind {
; RV64I-LABEL: lw:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lw a2, 4(a0)
; RV64I-NEXT:  lw zero, 0(a0)
; RV64I-NEXT:  sd a2, 0(a1)
; RV64I-NEXT:  ret
%1 = getelementptr i32, ptr %a, i64 1
%2 = load i32, ptr %1
%3 = sext i32 %2 to i64
; the unused load
will produce an anyext for selection
%4 = load volatile i32, ptr %a
store i64 %3, ptr %b
ret void
}

```

```

define void @lwu(ptr %a, ptr %b) nounwind {
; RV64I-LABEL: lwu:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lwu a0, 4(a0)
; RV64I-NEXT:  sd a0, 0(a1)
; RV64I-NEXT:  ret
%1 = getelementptr i32, ptr %a, i64 1
%2 = load i32, ptr %1
%3 = zext i32 %2 to i64
store i64 %3, ptr %b
ret void
}

```

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

\$OpenBSD: COPYRIGHT,v 1.3 2003/06/02 20:18:36 millert Exp \$

Copyright 1992, 1993, 1994 Henry Spencer. All rights reserved.  
This software is not subject to any license of the American Telephone and Telegraph Company or of the Regents of the University of California.

Permission is granted to anyone to use this software for any purpose on any computer system, and to alter it and redistribute it, subject to the following restrictions:

1. The author is not responsible for the consequences of use of this software, no matter how awful, even if they arise from flaws in it.
2. The origin of this software must not be misrepresented, either by explicit claim or by omission. Since few users ever read sources, credits must appear in the documentation.
3. Altered versions must be plainly marked as such, and must not be misrepresented as being the original software. Since few users ever read sources, credits must appear in the documentation.
4. This notice may not be removed or altered.

=====  
/\*-

\* Copyright (c) 1994

\* The Regents of the University of California. All rights reserved.

```

*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. Neither the name of the University nor the names of its contributors
*   may be used to endorse or promote products derived from this software
*   without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND
* ANY EXPRESS OR IMPLIED WARRANTIES,
* INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* @(#)COPYRIGHT 8.1 (Berkeley) 3/16/94
*/
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mcpu=skylake-avx512 -mattr=prefer-256-bit | FileCheck %s
--check-prefixes=CHECK,CHECK-SKX
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mcpu=skylake-avx512 -mattr=prefer-256-bit,avx512vbmi |
FileCheck %s --check-prefixes=CHECK,CHECK-SKX,CHECK-SKX-VBMI
; Make sure CPUs default to prefer-256-bit. avx512vnni isn't interesting as it just adds an isel peephole for
vpmaddwd+vpadd
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mcpu=skylake-avx512 | FileCheck %s --check-
prefixes=CHECK,CHECK-AVX512
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mattr=-avx512vnni -mcpu=cascadelake | FileCheck %s --
check-prefixes=CHECK,CHECK-AVX512
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mattr=-avx512vnni -mcpu=cooperlake | FileCheck %s --
check-prefixes=CHECK,CHECK-AVX512
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mcpu=cannonlake | FileCheck %s --check-
prefixes=CHECK,CHECK-VBMI
;
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mattr=-avx512vnni -mcpu=icelake-client | FileCheck %s --
check-prefixes=CHECK,CHECK-VBMI
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mattr=-avx512vnni -mcpu=icelake-server | FileCheck %s --
check-prefixes=CHECK,CHECK-VBMI

```

```
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mattr=-avx512vnni -mcpu=tigerlake | FileCheck %s --check-  
prefixes=CHECK,CHECK-VBMM
```

; This file primarily contains tests for specific places in X86ISelLowering.cpp that needed be made aware of the  
legalizer not allowing 512-bit vectors due to prefer-256-bit even though AVX512 is enabled.

```
define dso_local void @add256(<16 x i32>* %a, <16 x i32>* %b, <16 x i32>* %c) "min-legal-vector-  
width"="256" {  
; CHECK-LABEL: add256:  
; CHECK:      # %bb.0:  
; CHECK-NEXT: vmovdqa (%rdi), %ymm0  
; CHECK-NEXT: vmovdqa 32(%rdi), %ymm1  
; CHECK-NEXT: vpaddd 32(%rsi), %ymm1, %ymm1  
; CHECK-NEXT: vpaddd (%rsi), %ymm0, %ymm0  
; CHECK-NEXT: vmovdqa %ymm0, (%rdx)  
; CHECK-NEXT:  
    vmovdqa %ymm1, 32(%rdx)  
; CHECK-NEXT: vzeroupper  
; CHECK-NEXT: retq  
%d = load <16 x i32>, <16 x i32>* %a  
%e = load <16 x i32>, <16 x i32>* %b  
%f = add <16 x i32> %d, %e  
store <16 x i32> %f, <16 x i32>* %c  
ret void  
}
```

```
define dso_local void @add512(<16 x i32>* %a, <16 x i32>* %b, <16 x i32>* %c) "min-legal-vector-  
width"="512" {  
; CHECK-LABEL: add512:  
; CHECK:      # %bb.0:  
; CHECK-NEXT: vmovdqa64 (%rdi), %zmm0  
; CHECK-NEXT: vpaddd (%rsi), %zmm0, %zmm0  
; CHECK-NEXT: vmovdqa64 %zmm0, (%rdx)  
; CHECK-NEXT: vzeroupper  
; CHECK-NEXT: retq  
%d = load <16 x i32>, <16 x i32>* %a  
%e = load <16 x i32>, <16 x i32>* %b  
%f = add <16 x i32> %d, %e  
store <16 x i32> %f, <16 x i32>* %c  
ret void  
}
```

```
define dso_local void @avg_v64i8_256(<64 x i8>* %a, <64 x i8>* %b) "min-legal-vector-width"="256" {  
; CHECK-LABEL: avg_v64i8_256:  
; CHECK:      # %bb.0:  
; CHECK-NEXT: vmovdqa (%rsi), %ymm0  
; CHECK-NEXT: vmovdqa 32(%rsi), %ymm1  
; CHECK-NEXT: vpavgb (%rdi),
```



[illegible]

```

%8 = trunc <64 x i32> %7 to <64 x i8>
store <64 x i8> %8, <64 x i8>* undef,
align 4
ret void
}

```

```

define dso_local void @pmaddwd_32_256(<32 x i16>* %APtr, <32 x i16>* %BPtr, <16 x i32>* %CPtr) "min-
legal-vector-width"="256" {
; CHECK-LABEL: pmaddwd_32_256:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rdi), %ymm0
; CHECK-NEXT: vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT: vpmaddwd 32(%rsi), %ymm1, %ymm1
; CHECK-NEXT: vpmaddwd (%rsi), %ymm0, %ymm0
; CHECK-NEXT: vmovdqa %ymm0, (%rdx)
; CHECK-NEXT: vmovdqa %ymm1, 32(%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%A = load <32 x i16>, <32 x i16>* %APtr
%B = load <32 x i16>, <32 x i16>* %BPtr
%a = sext <32 x i16> %A to <32 x i32>
%b = sext <32 x i16> %B to <32 x i32>
%m = mul nsw <32 x i32> %a, %b
%odd = shufflevector <32 x i32> %m, <32 x i32> undef, <16 x i32> <i32 0, i32 2, i32 4, i32 6, i32 8, i32 10, i32
12, i32 14, i32 16, i32 18, i32 20, i32 22, i32 24, i32 26, i32 28, i32 30>
%even = shufflevector <32 x i32> %m, <32 x i32> undef, <16 x i32> <i32 1, i32 3, i32 5, i32
7, i32 9, i32 11, i32 13, i32 15, i32 17, i32 19, i32 21, i32 23, i32 25, i32 27, i32 29, i32 31>
%ret = add <16 x i32> %odd, %even
store <16 x i32> %ret, <16 x i32>* %CPtr
ret void
}

```

```

define dso_local void @pmaddwd_32_512(<32 x i16>* %APtr, <32 x i16>* %BPtr, <16 x i32>* %CPtr) "min-
legal-vector-width"="512" {
; CHECK-LABEL: pmaddwd_32_512:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa64 (%rdi), %zmm0
; CHECK-NEXT: vpmaddwd (%rsi), %zmm0, %zmm0
; CHECK-NEXT: vmovdqa64 %zmm0, (%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%A = load <32 x i16>, <32 x i16>* %APtr
%B = load <32 x i16>, <32 x i16>* %BPtr
%a = sext <32 x i16> %A to <32 x i32>
%b = sext <32 x i16> %B to <32 x i32>
%m = mul nsw <32 x i32> %a, %b
%odd = shufflevector <32 x i32> %m, <32 x i32> undef, <16 x i32> <i32 0, i32 2, i32 4, i32 6, i32 8, i32 10, i32
12, i32 14, i32 16, i32 18, i32 20, i32 22, i32 24, i32 26, i32 28, i32 30>

```

```

%even = shufflevector <32 x i32> %m, <32 x i32>
undef, <16 x i32> <i32 1, i32 3, i32 5, i32 7, i32 9, i32 11, i32 13, i32 15, i32 17, i32 19, i32 21, i32 23, i32 25, i32
27, i32 29, i32 31>
%ret = add <16 x i32> %odd, %even
store <16 x i32> %ret, <16 x i32>* %CPtr
ret void
}

define dso_local void @psubus_64i8_max_256(<64 x i8>* %xptr, <64 x i8>* %yptr, <64 x i8>* %zptr) "min-legal-
vector-width"="256" {
; CHECK-LABEL: psubus_64i8_max_256:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rdi), %ymm0
; CHECK-NEXT: vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT: vpsubusb 32(%rsi), %ymm1, %ymm1
; CHECK-NEXT: vpsubusb (%rsi), %ymm0, %ymm0
; CHECK-NEXT: vmovdqa %ymm0, (%rdx)
; CHECK-NEXT: vmovdqa %ymm1, 32(%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%x = load <64 x i8>, <64 x i8>* %xptr
%y = load <64 x i8>, <64 x i8>* %yptr
%cmp = icmp ult <64 x i8> %x, %y
%max = select <64 x i1> %cmp, <64 x i8> %y, <64 x i8> %x
%res = sub <64 x i8> %max, %y
store <64 x i8> %res, <64 x i8>* %zptr
ret void
}

define
dso_local void @psubus_64i8_max_512(<64 x i8>* %xptr, <64 x i8>* %yptr, <64 x i8>* %zptr) "min-legal-
vector-width"="512" {
; CHECK-LABEL: psubus_64i8_max_512:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa64 (%rdi), %zmm0
; CHECK-NEXT: vpsubusb (%rsi), %zmm0, %zmm0
; CHECK-NEXT: vmovdqa64 %zmm0, (%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%x = load <64 x i8>, <64 x i8>* %xptr
%y = load <64 x i8>, <64 x i8>* %yptr
%cmp = icmp ult <64 x i8> %x, %y
%max = select <64 x i1> %cmp, <64 x i8> %y, <64 x i8> %x
%res = sub <64 x i8> %max, %y
store <64 x i8> %res, <64 x i8>* %zptr
ret void
}

```

```

define dso_local i32 @_Z9test_charPcS_i_256(i8* nocapture readonly, i8* nocapture readonly, i32) "min-legal-
vector-width"="256" {
; CHECK-SKX-LABEL: _Z9test_charPcS_i_256:
; CHECK-SKX:      # %bb.0: # %entry
; CHECK-SKX-NEXT:  movl %edx, %eax
; CHECK-SKX-NEXT:  vpxor %xmm0, %xmm0, %xmm0
; CHECK-SKX-NEXT:  xorl %ecx, %ecx
; CHECK-SKX-NEXT:  vpxor %xmm1, %xmm1, %xmm1
;
; CHECK-SKX-NEXT:  vpxor %xmm2, %xmm2, %xmm2
; CHECK-SKX-NEXT:  .p2align 4, 0x90
; CHECK-SKX-NEXT:  .LBB8_1: # %vector.body
; CHECK-SKX-NEXT:  # =>This Inner Loop Header: Depth=1
; CHECK-SKX-NEXT:  vpmovsxbw 16(%rdi,%rcx), %ymm3
; CHECK-SKX-NEXT:  vpmovsxbw (%rdi,%rcx), %ymm4
; CHECK-SKX-NEXT:  vpmovsxbw 16(%rsi,%rcx), %ymm5
; CHECK-SKX-NEXT:  vpmaddwd %ymm3, %ymm5, %ymm3
; CHECK-SKX-NEXT:  vpadd %ymm2, %ymm3, %ymm2
; CHECK-SKX-NEXT:  vpmovsxbw (%rsi,%rcx), %ymm3
; CHECK-SKX-NEXT:  vpmaddwd %ymm4, %ymm3, %ymm3
; CHECK-SKX-NEXT:  vpadd %ymm1, %ymm3, %ymm1
; CHECK-SKX-NEXT:  addq $32, %rcx
; CHECK-SKX-NEXT:  cmpq %rcx, %rax
; CHECK-SKX-NEXT:  jne .LBB8_1
; CHECK-SKX-NEXT:  # %bb.2: # %middle.block
; CHECK-SKX-NEXT:  vpadd %ymm0, %ymm1, %ymm1
; CHECK-SKX-NEXT:  vpadd %ymm0, %ymm2, %ymm0
; CHECK-SKX-NEXT:  vpadd %ymm0, %ymm1, %ymm0
; CHECK-SKX-NEXT:  vextracti128 $1, %ymm0, %xmm1
; CHECK-SKX-NEXT:  vpadd %xmm1, %xmm0, %xmm0
; CHECK-SKX-NEXT:
;     vpsltd {{.*#+}} xmm1 = xmm0[2,3,2,3]
; CHECK-SKX-NEXT:  vpadd %xmm1, %xmm0, %xmm0
; CHECK-SKX-NEXT:  vpsrlq $32, %xmm0, %xmm1
; CHECK-SKX-NEXT:  vpadd %xmm1, %xmm0, %xmm0
; CHECK-SKX-NEXT:  vmovd %xmm0, %eax
; CHECK-SKX-NEXT:  vzeroupper
; CHECK-SKX-NEXT:  retq
;
; CHECK-AVX512-LABEL: _Z9test_charPcS_i_256:
; CHECK-AVX512:    # %bb.0: # %entry
; CHECK-AVX512-NEXT:  movl %edx, %eax
; CHECK-AVX512-NEXT:  vpxor %xmm0, %xmm0, %xmm0
; CHECK-AVX512-NEXT:  xorl %ecx, %ecx
; CHECK-AVX512-NEXT:  vpxor %xmm1, %xmm1, %xmm1
; CHECK-AVX512-NEXT:  vpxor %xmm2, %xmm2, %xmm2
; CHECK-AVX512-NEXT:  .p2align 4, 0x90
; CHECK-AVX512-NEXT:  .LBB8_1: # %vector.body

```

```

; CHECK-AVX512-NEXT: # =>This Inner Loop Header: Depth=1
; CHECK-AVX512-NEXT: vpmovsxbw 16(%rdi,%rcx), %ymm3
; CHECK-AVX512-NEXT: vpmovsxbw (%rdi,%rcx), %ymm4
; CHECK-AVX512-NEXT: vpmovsxbw 16(%rsi,%rcx), %ymm5
; CHECK-AVX512-NEXT: vpmaddwd %ymm3, %ymm5, %ymm3
; CHECK-AVX512-NEXT: vpaddd %ymm2, %ymm3,
%ymm2
; CHECK-AVX512-NEXT: vpmovsxbw (%rsi,%rcx), %ymm3
; CHECK-AVX512-NEXT: vpmaddwd %ymm4, %ymm3, %ymm3
; CHECK-AVX512-NEXT: vpaddd %ymm1, %ymm3, %ymm1
; CHECK-AVX512-NEXT: addq $32, %rcx
; CHECK-AVX512-NEXT: cmpq %rcx, %rax
; CHECK-AVX512-NEXT: jne .LBB8_1
; CHECK-AVX512-NEXT: # %bb.2: # %middle.block
; CHECK-AVX512-NEXT: vpaddd %ymm0, %ymm1, %ymm1
; CHECK-AVX512-NEXT: vpaddd %ymm0, %ymm2, %ymm0
; CHECK-AVX512-NEXT: vpaddd %ymm0, %ymm1, %ymm0
; CHECK-AVX512-NEXT: vextracti128 $1, %ymm0, %xmm1
; CHECK-AVX512-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-AVX512-NEXT: vpslufd {{.*#+}} xmm1 = xmm0[2,3,2,3]
; CHECK-AVX512-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-AVX512-NEXT: vpsrlq $32, %xmm0, %xmm1
; CHECK-AVX512-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-AVX512-NEXT: vmovd %xmm0, %eax
; CHECK-AVX512-NEXT: vzeroupper
; CHECK-AVX512-NEXT: retq
;
; CHECK-VBMI-LABEL: _Z9test_charPcS_i_256:
; CHECK-VBMI: # %bb.0: # %entry
; CHECK-VBMI-NEXT:
movl %edx, %eax
; CHECK-VBMI-NEXT: vpxor %xmm0, %xmm0, %xmm0
; CHECK-VBMI-NEXT: xorl %ecx, %ecx
; CHECK-VBMI-NEXT: vpxor %xmm1, %xmm1, %xmm1
; CHECK-VBMI-NEXT: vpxor %xmm2, %xmm2, %xmm2
; CHECK-VBMI-NEXT: .p2align 4, 0x90
; CHECK-VBMI-NEXT: .LBB8_1: # %vector.body
; CHECK-VBMI-NEXT: # =>This Inner Loop Header: Depth=1
; CHECK-VBMI-NEXT: vpmovsxbw 16(%rdi,%rcx), %ymm3
; CHECK-VBMI-NEXT: vpmovsxbw (%rdi,%rcx), %ymm4
; CHECK-VBMI-NEXT: vpmovsxbw 16(%rsi,%rcx), %ymm5
; CHECK-VBMI-NEXT: vpmaddwd %ymm3, %ymm5, %ymm3
; CHECK-VBMI-NEXT: vpaddd %ymm2, %ymm3, %ymm2
; CHECK-VBMI-NEXT: vpmovsxbw (%rsi,%rcx), %ymm3
; CHECK-VBMI-NEXT: vpmaddwd %ymm4, %ymm3, %ymm3
; CHECK-VBMI-NEXT: vpaddd %ymm1, %ymm3, %ymm1
; CHECK-VBMI-NEXT: addq $32, %rcx
; CHECK-VBMI-NEXT: cmpq %rcx, %rax

```

```
; CHECK-VBMI-NEXT:  jne .LBB8_1
; CHECK-VBMI-NEXT:  # %bb.2: # %middle.block
; CHECK-VBMI-NEXT:  vpaddd %ymm0, %ymm1, %ymm1
; CHECK-VBMI-NEXT:  vpaddd %ymm0, %ymm2,
    %ymm0
; CHECK-VBMI-NEXT:  vpaddd %ymm0, %ymm1, %ymm0
; CHECK-VBMI-NEXT:  vextracti128 $1, %ymm0, %xmm1
; CHECK-VBMI-NEXT:  vpaddd %xmm1, %xmm0, %xmm0
; CHECK-VBMI-NEXT:  vpslufd {{.*#+}} xmm1 = xmm0[2,3,2,3]
; CHECK-VBMI-NEXT:  vpaddd %xmm1, %xmm0, %xmm0
; CHECK-VBMI-NEXT:  vpslufd {{.*#+}} xmm1 = xmm0[1,1,1,1]
; CHECK-VBMI-NEXT:  vpaddd %xmm1, %xmm0, %xmm0
; CHECK-VBMI-NEXT:  vmovd %xmm0, %eax
; CHECK-VBMI-NEXT:  vzeroupper
; CHECK-VBMI-NEXT:  retq
```

entry:

```
%3 = zext i32 %2 to i64
```

```
br label %vector.body
```

vector.body:

```
%index = phi i64 [ %index.next, %vector.body ], [ 0, %entry ]
%vec.phi = phi <32 x i32> [ %11, %vector.body ], [ zeroinitializer, %entry ]
%4 = getelementptr inbounds i8, i8* %0, i64 %index
%5 = bitcast i8* %4 to <32 x i8*>
%wide.load = load <32 x i8>, <32 x i8*> %5, align 1
%6 = sext <32 x i8> %wide.load to <32 x i32>
%7 = getelementptr inbounds i8, i8* %1, i64 %index
%8 = bitcast i8* %7 to <32 x i8*>
%wide.load14 = load <32 x
i8>, <32 x i8*> %8, align 1
%9 = sext <32 x i8> %wide.load14 to <32 x i32>
%10 = mul nsw <32 x i32> %9, %6
%11 = add nsw <32 x i32> %10, %vec.phi
%index.next = add i64 %index, 32
%12 = icmp eq i64 %index.next, %3
br i1 %12, label %middle.block, label %vector.body
```

middle.block:

```
%rdx.shuf1 = shufflevector <32 x i32> %11, <32 x i32> undef, <32 x i32> <i32 16, i32 17, i32 18, i32 19, i32 20,
i32 21, i32 22, i32 23, i32 24, i32 25, i32 26, i32 27, i32 28, i32 29, i32 30, i32 31, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef>
%bin.rdx1 = add <32 x i32> %11, %rdx.shuf1
%rdx.shuf = shufflevector <32 x i32> %bin.rdx1, <32 x i32> undef, <32 x i32> <i32 8, i32 9, i32 10, i32 11, i32
12, i32 13, i32 14, i32 15, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef>
```



```

; CHECK-SKX-NEXT:  vpsrlq $32, %xmm0, %xmm1
; CHECK-SKX-NEXT:  vpaddq %xmm1, %xmm0, %xmm0
; CHECK-SKX-NEXT:
    vmovd %xmm0, %eax
; CHECK-SKX-NEXT:  vzeroupper
; CHECK-SKX-NEXT:  retq
;
; CHECK-AVX512-LABEL: _Z9test_charPcS_i_512:
; CHECK-AVX512:    # %bb.0: # %entry
; CHECK-AVX512-NEXT:  movl %edx, %eax
; CHECK-AVX512-NEXT:  vpxor %xmm0, %xmm0, %xmm0
; CHECK-AVX512-NEXT:  xorl %ecx, %ecx
; CHECK-AVX512-NEXT:  vpxor %xmm1, %xmm1, %xmm1
; CHECK-AVX512-NEXT:  .p2align 4, 0x90
; CHECK-AVX512-NEXT: .LBB9_1: # %vector.body
; CHECK-AVX512-NEXT: # =>This Inner Loop Header: Depth=1
; CHECK-AVX512-NEXT: vpmovsxbw (%rdi,%rcx), %zmm2
; CHECK-AVX512-NEXT: vpmovsxbw (%rsi,%rcx), %zmm3
; CHECK-AVX512-NEXT: vpmaddwd %zmm2, %zmm3, %zmm2
; CHECK-AVX512-NEXT: vpaddq %zmm1, %zmm2, %zmm1
; CHECK-AVX512-NEXT: addq $32, %rcx
; CHECK-AVX512-NEXT: cmpq %rcx, %rax
; CHECK-AVX512-NEXT: jne .LBB9_1
; CHECK-AVX512-NEXT: # %bb.2: # %middle.block
; CHECK-AVX512-NEXT: vpaddq %zmm0, %zmm1, %zmm0
; CHECK-AVX512-NEXT: vextracti64x4 $1, %zmm0, %ymm1
; CHECK-AVX512-NEXT: vpaddq
    %zmm1, %zmm0, %zmm0
; CHECK-AVX512-NEXT: vextracti128 $1, %ymm0, %xmm1
; CHECK-AVX512-NEXT: vpaddq %xmm1, %xmm0, %xmm0
; CHECK-AVX512-NEXT: vpshufd {{.*#+}} xmm1 = xmm0[2,3,2,3]
; CHECK-AVX512-NEXT: vpaddq %xmm1, %xmm0, %xmm0
; CHECK-AVX512-NEXT: vpsrlq $32, %xmm0, %xmm1
; CHECK-AVX512-NEXT: vpaddq %xmm1, %xmm0, %xmm0
; CHECK-AVX512-NEXT: vmovd %xmm0, %eax
; CHECK-AVX512-NEXT: vzeroupper
; CHECK-AVX512-NEXT: retq
;
; CHECK-VBMI-LABEL: _Z9test_charPcS_i_512:
; CHECK-VBMI:    # %bb.0: # %entry
; CHECK-VBMI-NEXT:  movl %edx, %eax
; CHECK-VBMI-NEXT:  vpxor %xmm0, %xmm0, %xmm0
; CHECK-VBMI-NEXT:  xorl %ecx, %ecx
; CHECK-VBMI-NEXT:  vpxor %xmm1, %xmm1, %xmm1
; CHECK-VBMI-NEXT:  .p2align 4, 0x90
; CHECK-VBMI-NEXT: .LBB9_1: # %vector.body
; CHECK-VBMI-NEXT: # =>This Inner Loop Header: Depth=1
; CHECK-VBMI-NEXT: vpmovsxbw (%rdi,%rcx), %zmm2

```



```

; CHECK-VBMI-NEXT: vpmovsxbw (%rsi,%rcx), %zmm3
; CHECK-VBMI-NEXT: vpmaddwd %zmm2, %zmm3, %zmm2
;
CHECK-VBMI-NEXT: vpaddd %zmm1, %zmm2, %zmm1
; CHECK-VBMI-NEXT: addq $32, %rcx
; CHECK-VBMI-NEXT: cmpq %rcx, %rax
; CHECK-VBMI-NEXT: jne .LBB9_1
; CHECK-VBMI-NEXT: # %bb.2: # %middle.block
; CHECK-VBMI-NEXT: vpaddd %zmm0, %zmm1, %zmm0
; CHECK-VBMI-NEXT: vextracti64x4 $1, %zmm0, %ymm1
; CHECK-VBMI-NEXT: vpaddd %zmm1, %zmm0, %zmm0
; CHECK-VBMI-NEXT: vextracti128 $1, %ymm0, %xmm1
; CHECK-VBMI-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-VBMI-NEXT: vpshufd {{.*#+}} xmm1 = xmm0[2,3,2,3]
; CHECK-VBMI-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-VBMI-NEXT: vpshufd {{.*#+}} xmm1 = xmm0[1,1,1,1]
; CHECK-VBMI-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-VBMI-NEXT: vmovd %xmm0, %eax
; CHECK-VBMI-NEXT: vzeroupper
; CHECK-VBMI-NEXT: retq
entry:
%3 = zext i32 %2 to i64
br label %vector.body

```

vector.body:

```

%index = phi i64 [ %index.next, %vector.body ], [ 0, %entry ]
%vec.phi = phi <32 x i32> [ %11, %vector.body ], [ zeroinitializer, %entry ]
%4 = getelementptr
inbounds i8, i8* %0, i64 %index
%5 = bitcast i8* %4 to <32 x i8>*
%wide.load = load <32 x i8>, <32 x i8>* %5, align 1
%6 = sext <32 x i8> %wide.load to <32 x i32>
%7 = getelementptr inbounds i8, i8* %1, i64 %index
%8 = bitcast i8* %7 to <32 x i8>*
%wide.load14 = load <32 x i8>, <32 x i8>* %8, align 1
%9 = sext <32 x i8> %wide.load14 to <32 x i32>
%10 = mul nsw <32 x i32> %9, %6
%11 = add nsw <32 x i32> %10, %vec.phi
%index.next = add i64 %index, 32
%12 = icmp eq i64 %index.next, %3
br i1 %12, label %middle.block, label %vector.body

```

middle.block:

```

%rdx.shuf1 = shufflevector <32 x i32> %11, <32 x i32> undef, <32 x i32> <i32 16, i32 17, i32 18, i32 19, i32 20,
i32 21, i32 22, i32 23, i32 24, i32 25, i32 26, i32 27, i32 28, i32 29, i32 30, i32 31, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx1 = add <32 x i32> %11, %rdx.shuf1

```

```

%rdx.shuf = shufflevector <32 x i32> %bin.rdx1, <32 x i32> undef, <32 x i32> <i32 8, i32 9, i32 10, i32 11, i32
12, i32 13, i32 14, i32 15, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx = add <32 x i32> %bin.rdx1, %rdx.shuf
%rdx.shuf15 = shufflevector <32 x i32> %bin.rdx, <32 x i32> undef, <32 x i32> <i32 4, i32 5, i32 6, i32 7, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx32 = add <32 x i32> %bin.rdx, %rdx.shuf15
%rdx.shuf17 = shufflevector <32 x i32> %bin.rdx32, <32 x i32> undef,
<32 x i32> <i32 2, i32 3, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef>
%bin.rdx18 = add <32 x i32> %bin.rdx32, %rdx.shuf17
%rdx.shuf19 = shufflevector <32 x i32> %bin.rdx18, <32 x i32> undef, <32 x i32> <i32 1, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx20 = add <32 x i32> %bin.rdx18, %rdx.shuf19
%13 = extractelement <32 x i32> %bin.rdx20, i32 0
ret i32 %13
}

```

```

@a = dso_local global [1024 x i8]
zeroinitializer, align 16
@b = dso_local global [1024 x i8] zeroinitializer, align 16

```

```

define dso_local i32 @sad_16i8_256() "min-legal-vector-width"="256" {
; CHECK-SKX-LABEL: sad_16i8_256:
; CHECK-SKX:      # %bb.0: # %entry
; CHECK-SKX-NEXT: vpxor %xmm0, %xmm0, %xmm0
; CHECK-SKX-NEXT: movq $-1024, %rax # imm = 0xFC00
; CHECK-SKX-NEXT: vpxor %xmm1, %xmm1, %xmm1
; CHECK-SKX-NEXT: .p2align 4, 0x90
; CHECK-SKX-NEXT: .LBB10_1: # %vector.body
; CHECK-SKX-NEXT: # =>This Inner Loop Header: Depth=1
; CHECK-SKX-NEXT: vmovdqu a+1024(%rax), %xmm2
; CHECK-SKX-NEXT: vpsadbw b+1024(%rax), %xmm2, %xmm2
; CHECK-SKX-NEXT: vpadd %ymm1, %ymm2, %ymm1
; CHECK-SKX-NEXT: addq $4, %rax
; CHECK-SKX-NEXT: jne .LBB10_1
; CHECK-SKX-NEXT: # %bb.2: # %middle.block
; CHECK-SKX-NEXT: vpadd %ymm0, %ymm1, %ymm0
; CHECK-SKX-NEXT: vextracti128 $1, %ymm0, %xmm1
; CHECK-SKX-NEXT: vpadd %xmm1, %xmm0, %xmm0
; CHECK-SKX-NEXT: vpshufd {{.*#+}} xmm1 = xmm0[2,3,2,3]

```

```

; CHECK-SKX-NEXT:
    vpaddd %xmm1, %xmm0, %xmm0
; CHECK-SKX-NEXT:  vpsrlq $32, %xmm0, %xmm1
; CHECK-SKX-NEXT:  vpaddd %xmm1, %xmm0, %xmm0
; CHECK-SKX-NEXT:  vmovd %xmm0, %eax
; CHECK-SKX-NEXT:  vzeroupper
; CHECK-SKX-NEXT:  retq
;
; CHECK-AVX512-LABEL: sad_16i8_256:
; CHECK-AVX512:    # %bb.0: # %entry
; CHECK-AVX512-NEXT:  vpxor %xmm0, %xmm0, %xmm0
; CHECK-AVX512-NEXT:  movq $-1024, %rax # imm = 0xFC00
; CHECK-AVX512-NEXT:  vpxor %xmm1, %xmm1, %xmm1
; CHECK-AVX512-NEXT:  .p2align 4, 0x90
; CHECK-AVX512-NEXT: .LBB10_1: # %vector.body
; CHECK-AVX512-NEXT: # =>This Inner Loop Header: Depth=1
; CHECK-AVX512-NEXT: vmovdqu a+1024(%rax), %xmm2
; CHECK-AVX512-NEXT: vpsadbw b+1024(%rax), %xmm2, %xmm2
; CHECK-AVX512-NEXT: vpaddd %ymm1, %ymm2, %ymm1
; CHECK-AVX512-NEXT: addq $4, %rax
; CHECK-AVX512-NEXT: jne .LBB10_1
; CHECK-AVX512-NEXT: # %bb.2: # %middle.block
; CHECK-AVX512-NEXT: vpaddd %ymm0, %ymm1, %ymm0
; CHECK-AVX512-NEXT: vextracti128 $1, %ymm0, %xmm1
;
CHECK-AVX512-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-AVX512-NEXT: vpshufd {{.*#+}} xmm1 = xmm0[2,3,2,3]
; CHECK-AVX512-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-AVX512-NEXT: vpsrlq $32, %xmm0, %xmm1
; CHECK-AVX512-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-AVX512-NEXT: vmovd %xmm0, %eax
; CHECK-AVX512-NEXT: vzeroupper
; CHECK-AVX512-NEXT: retq
;
; CHECK-VBMI-LABEL: sad_16i8_256:
; CHECK-VBMI:    # %bb.0: # %entry
; CHECK-VBMI-NEXT:  vpxor %xmm0, %xmm0, %xmm0
; CHECK-VBMI-NEXT:  movq $-1024, %rax # imm = 0xFC00
; CHECK-VBMI-NEXT:  vpxor %xmm1, %xmm1, %xmm1
; CHECK-VBMI-NEXT:  .p2align 4, 0x90
; CHECK-VBMI-NEXT: .LBB10_1: # %vector.body
; CHECK-VBMI-NEXT: # =>This Inner Loop Header: Depth=1
; CHECK-VBMI-NEXT: vmovdqu a+1024(%rax), %xmm2
; CHECK-VBMI-NEXT: vpsadbw b+1024(%rax), %xmm2, %xmm2
; CHECK-VBMI-NEXT: vpaddd %ymm1, %ymm2, %ymm1
; CHECK-VBMI-NEXT: addq $4, %rax
; CHECK-VBMI-NEXT: jne .LBB10_1
; CHECK-VBMI-NEXT: # %bb.2: # %middle.block

```

```

;
CHECK-VBMI-NEXT:  vpaddd %ymm0, %ymm1, %ymm0
; CHECK-VBMI-NEXT:  vextracti128 $1, %ymm0, %xmm1
; CHECK-VBMI-NEXT:  vpaddd %xmm1, %xmm0, %xmm0
; CHECK-VBMI-NEXT:  vpshufd {{.*#+}} xmm1 = xmm0[2,3,2,3]
; CHECK-VBMI-NEXT:  vpaddd %xmm1, %xmm0, %xmm0
; CHECK-VBMI-NEXT:  vpshufd {{.*#+}} xmm1 = xmm0[1,1,1,1]
; CHECK-VBMI-NEXT:  vpaddd %xmm1, %xmm0, %xmm0
; CHECK-VBMI-NEXT:  vmovd %xmm0, %eax
; CHECK-VBMI-NEXT:  vzeroupper
; CHECK-VBMI-NEXT:  retq
entry:
br label %vector.body

vector.body:
%index = phi i64 [ 0, %entry ], [ %index.next, %vector.body ]
%vec.phi = phi <16 x i32> [ zeroinitializer, %entry ], [ %10, %vector.body ]
%0 = getelementptr inbounds [1024 x i8], [1024 x i8]* @a, i64 0, i64 %index
%1 = bitcast i8* %0 to <16 x i8>*
%wide.load = load <16 x i8>, <16 x i8>* %1, align 4
%2 = zext <16 x i8> %wide.load to <16 x i32>
%3 = getelementptr inbounds [1024 x i8], [1024 x i8]* @b, i64 0, i64 %index
%4 = bitcast i8* %3 to <16 x i8>*
%wide.load1
= load <16 x i8>, <16 x i8>* %4, align 4
%5 = zext <16 x i8> %wide.load1 to <16 x i32>
%6 = sub nsw <16 x i32> %2, %5
%7 = icmp sgt <16 x i32> %6, <i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1>
%8 = sub nsw <16 x i32> zeroinitializer, %6
%9 = select <16 x i1> %7, <16 x i32> %6, <16 x i32> %8
%10 = add nsw <16 x i32> %9, %vec.phi
%index.next = add i64 %index, 4
%11 = icmp eq i64 %index.next, 1024
br i1 %11, label %middle.block, label %vector.body

middle.block:
%rdx.shuf = shufflevector <16 x i32> %10, <16 x i32> undef, <16 x i32> <i32 8, i32 9, i32 10, i32 11, i32 12, i32 13, i32 14, i32 15, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx = add <16 x i32> %10, %rdx.shuf
%rdx.shuf2 = shufflevector <16 x i32> %bin.rdx, <16 x i32> undef, <16 x i32> <i32 4, i32 5, i32 6, i32 7, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx2 = add <16 x i32> %bin.rdx, %rdx.shuf2
%rdx.shuf3 = shufflevector <16 x i32> %bin.rdx2, <16 x i32> undef, <16 x i32> <i32 2, i32 3, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx3 = add <16 x i32> %bin.rdx2, %rdx.shuf3

```

```

%rdx.shuf4 = shufflevector <16 x i32> %bin.rdx3, <16 x i32> undef, <16 x i32> <i32 1, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx4 = add <16 x i32> %bin.rdx3, %rdx.shuf4
%12 = extractelement <16 x i32> %bin.rdx4, i32 0
ret i32 %12
}

```

```

define dso_local i32 @sad_16i8_512() "min-legal-vector-width"="512" {
; CHECK-SKX-LABEL: sad_16i8_512:
; CHECK-SKX:      # %bb.0: # %entry
; CHECK-SKX-NEXT: vpxor %xmm0, %xmm0, %xmm0
; CHECK-SKX-NEXT:
    movq $-1024, %rax # imm = 0xFC00
; CHECK-SKX-NEXT: .p2align 4, 0x90
; CHECK-SKX-NEXT: .LBB11_1: # %vector.body
; CHECK-SKX-NEXT: # =>This Inner Loop Header: Depth=1
; CHECK-SKX-NEXT: vmovdqu a+1024(%rax), %xmm1
; CHECK-SKX-NEXT: vpsadbw b+1024(%rax), %xmm1, %xmm1
; CHECK-SKX-NEXT: vpadd %zmm0, %zmm1, %zmm0
; CHECK-SKX-NEXT: addq $4, %rax
; CHECK-SKX-NEXT: jne .LBB11_1
; CHECK-SKX-NEXT: # %bb.2: # %middle.block
; CHECK-SKX-NEXT: vextracti64x4 $1, %zmm0, %ymm1
; CHECK-SKX-NEXT: vpadd %zmm1, %zmm0, %zmm0
; CHECK-SKX-NEXT: vextracti128 $1, %ymm0, %xmm1
; CHECK-SKX-NEXT: vpadd %xmm1, %xmm0, %xmm0
; CHECK-SKX-NEXT: vpsltd {{.*#+}} xmm1 = xmm0[2,3,2,3]
; CHECK-SKX-NEXT: vpadd %xmm1, %xmm0, %xmm0
; CHECK-SKX-NEXT: vpsrlq $32, %xmm0, %xmm1
; CHECK-SKX-NEXT: vpadd %xmm1, %xmm0, %xmm0
; CHECK-SKX-NEXT: vmovd %xmm0, %eax
; CHECK-SKX-NEXT: vzeroupper
; CHECK-SKX-NEXT: retq
;
; CHECK-AVX512-LABEL: sad_16i8_512:
; CHECK-AVX512:
    # %bb.0: # %entry
; CHECK-AVX512-NEXT: vpxor %xmm0, %xmm0, %xmm0
; CHECK-AVX512-NEXT: movq $-1024, %rax # imm = 0xFC00
; CHECK-AVX512-NEXT: .p2align 4, 0x90
; CHECK-AVX512-NEXT: .LBB11_1: # %vector.body
; CHECK-AVX512-NEXT: # =>This Inner Loop Header: Depth=1
; CHECK-AVX512-NEXT: vmovdqu a+1024(%rax), %xmm1
; CHECK-AVX512-NEXT: vpsadbw b+1024(%rax), %xmm1, %xmm1
; CHECK-AVX512-NEXT: vpadd %zmm0, %zmm1, %zmm0
; CHECK-AVX512-NEXT: addq $4, %rax
; CHECK-AVX512-NEXT: jne .LBB11_1

```

```

; CHECK-AVX512-NEXT: # %bb.2: # %middle.block
; CHECK-AVX512-NEXT: vextracti64x4 $1, %zmm0, %ymm1
; CHECK-AVX512-NEXT: vpaddd %zmm1, %zmm0, %zmm0
; CHECK-AVX512-NEXT: vextracti128 $1, %ymm0, %xmm1
; CHECK-AVX512-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-AVX512-NEXT: vpshufd {{.*#+}} xmm1 = xmm0[2,3,2,3]
; CHECK-AVX512-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-AVX512-NEXT: vpsrlq $32, %xmm0, %xmm1
; CHECK-AVX512-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-AVX512-NEXT:
    vmovd %xmm0, %eax
; CHECK-AVX512-NEXT: vzeroupper
; CHECK-AVX512-NEXT: retq
;
; CHECK-VBMI-LABEL: sad_16i8_512:
; CHECK-VBMI: # %bb.0: # %entry
; CHECK-VBMI-NEXT: vpxor %xmm0, %xmm0, %xmm0
; CHECK-VBMI-NEXT: movq $-1024, %rax # imm = 0xFC00
; CHECK-VBMI-NEXT: .p2align 4, 0x90
; CHECK-VBMI-NEXT: .LBB11_1: # %vector.body
; CHECK-VBMI-NEXT: # =>This Inner Loop Header: Depth=1
; CHECK-VBMI-NEXT: vmovdqu a+1024(%rax), %xmm1
; CHECK-VBMI-NEXT: vpsadbw b+1024(%rax), %xmm1, %xmm1
; CHECK-VBMI-NEXT: vpaddd %zmm0, %zmm1, %zmm0
; CHECK-VBMI-NEXT: addq $4, %rax
; CHECK-VBMI-NEXT: jne .LBB11_1
; CHECK-VBMI-NEXT: # %bb.2: # %middle.block
; CHECK-VBMI-NEXT: vextracti64x4 $1, %zmm0, %ymm1
; CHECK-VBMI-NEXT: vpaddd %zmm1, %zmm0, %zmm0
; CHECK-VBMI-NEXT: vextracti128 $1, %ymm0, %xmm1
; CHECK-VBMI-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-VBMI-NEXT: vpshufd {{.*#+}} xmm1 = xmm0[2,3,2,3]
; CHECK-VBMI-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-VBMI-NEXT:
    vpshufd {{.*#+}} xmm1 = xmm0[1,1,1,1]
; CHECK-VBMI-NEXT: vpaddd %xmm1, %xmm0, %xmm0
; CHECK-VBMI-NEXT: vmovd %xmm0, %eax
; CHECK-VBMI-NEXT: vzeroupper
; CHECK-VBMI-NEXT: retq
entry:
br label %vector.body

vector.body:
%index = phi i64 [ 0, %entry ], [ %index.next, %vector.body ]
%vec.phi = phi <16 x i32> [ zeroinitializer, %entry ], [ %10, %vector.body ]
%0 = getelementptr inbounds [1024 x i8], [1024 x i8]* @a, i64 0, i64 %index
%1 = bitcast i8* %0 to <16 x i8>*
%wide.load = load <16 x i8>, <16 x i8>* %1, align 4

```

```

%2 = zext <16 x i8> %wide.load to <16 x i32>
%3 = getelementptr inbounds [1024 x i8], [1024 x i8]* @b, i64 0, i64 %index
%4 = bitcast i8* %3 to <16 x i8>*
%wide.load1 = load <16 x i8>, <16 x i8>* %4, align 4
%5 = zext <16 x i8> %wide.load1 to <16 x i32>
%6 = sub nsw <16 x i32> %2, %5
%7 = icmp sgt <16 x i32> %6, <i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1>
br i1 %11, label %middle.block, label %vector.body

```

middle.block:

```

%rdx.shuf = shufflevector <16 x i32> %10, <16 x i32> undef, <16 x i32> <i32 8, i32 9, i32 10, i32 11, i32 12, i32 13, i32 14, i32 15, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx = add <16 x i32> %10, %rdx.shuf
%rdx.shuf2 = shufflevector <16 x i32> %bin.rdx, <16 x i32> undef, <16 x i32> <i32 4, i32 5, i32 6, i32 7, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx2 = add <16 x i32> %bin.rdx, %rdx.shuf2
%rdx.shuf3 = shufflevector <16 x i32> %bin.rdx2, <16 x i32> undef, <16 x i32> <i32 2, i32 3, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx3 = add <16 x i32> %bin.rdx2, %rdx.shuf3
%rdx.shuf4 = shufflevector <16 x i32> %bin.rdx3, <16 x i32> undef, <16 x i32> <i32 1, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx4 = add <16 x i32> %bin.rdx3, %rdx.shuf4
%12 = extractelement <16 x i32> %bin.rdx4, i32 0
ret i32 %12
}

```

```

define dso_local void @sbto16f32_256(<16 x i16> %a, <16 x float>* %res) "min-legal-vector-width"="256" {
; CHECK-LABEL: sbto16f32_256:
; CHECK:      # %bb.0:
; CHECK-NEXT: vpmovw2m %ymm0, %k0
; CHECK-NEXT: kshiftrw $8, %k0, %k1
; CHECK-NEXT: vpmovm2d %k1, %ymm0
; CHECK-NEXT: vcvtqd2ps %ymm0, %ymm0
; CHECK-NEXT: vpmovm2d %k0, %ymm1
; CHECK-NEXT: vcvtqd2ps %ymm1, %ymm1
; CHECK-NEXT: vmovaps %ymm1, (%rdi)
; CHECK-NEXT: vmovaps %ymm0, 32(%rdi)
; CHECK-NEXT:

```

```

    vzeroupper
; CHECK-NEXT:    retq
%mask = icmp slt <16 x i16> %a, zeroinitializer
%1 = sitofp <16 x i1> %mask to <16 x float>
store <16 x float> %1, <16 x float>* %res
ret void
}

define dso_local void @sbto16f32_512(<16 x i16> %a, <16 x float>* %res) "min-legal-vector-width"="512" {
; CHECK-LABEL: sbto16f32_512:
; CHECK:        # %bb.0:
; CHECK-NEXT:    vpmovw2m %ymm0, %k0
; CHECK-NEXT:    vpmovm2d %k0, %zmm0
; CHECK-NEXT:    vcvtdq2ps %zmm0, %zmm0
; CHECK-NEXT:    vmovaps %zmm0, (%rdi)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
%mask = icmp slt <16 x i16> %a, zeroinitializer
%1 = sitofp <16 x i1> %mask to <16 x float>
store <16 x float> %1, <16 x float>* %res
ret void
}

define dso_local void @sbto16f64_256(<16 x i16> %a, <16 x double>* %res) "min-legal-vector-width"="256" {
; CHECK-LABEL: sbto16f64_256:
; CHECK:        # %bb.0:
; CHECK-NEXT:    vpmovw2m %ymm0, %k0
; CHECK-NEXT:    kshiftrw $8, %k0, %k1
; CHECK-NEXT:    vpmovm2d %k1, %ymm0
; CHECK-NEXT:
    vcvtdq2pd %xmm0, %ymm1
; CHECK-NEXT:    vextracti128 $1, %ymm0, %xmm0
; CHECK-NEXT:    vcvtdq2pd %xmm0, %ymm0
; CHECK-NEXT:    vpmovm2d %k0, %ymm2
; CHECK-NEXT:    vcvtdq2pd %xmm2, %ymm3
; CHECK-NEXT:    vextracti128 $1, %ymm2, %xmm2
; CHECK-NEXT:    vcvtdq2pd %xmm2, %ymm2
; CHECK-NEXT:    vmovaps %ymm2, 32(%rdi)
; CHECK-NEXT:    vmovaps %ymm3, (%rdi)
; CHECK-NEXT:    vmovaps %ymm0, 96(%rdi)
; CHECK-NEXT:    vmovaps %ymm1, 64(%rdi)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
%mask = icmp slt <16 x i16> %a, zeroinitializer
%1 = sitofp <16 x i1> %mask to <16 x double>
store <16 x double> %1, <16 x double>* %res
ret void
}

```



```

define dso_local void @sbto16f64_512(<16 x i16> %a, <16 x double>* %res) "min-legal-vector-width"="512" {
; CHECK-LABEL: sbto16f64_512:
; CHECK:      # %bb.0:
; CHECK-NEXT:  vpmovw2m %ymm0, %k0
; CHECK-NEXT:  vpmovm2d %k0, %zmm0
; CHECK-NEXT:  vcvtdq2pd %ymm0, %zmm1
; CHECK-NEXT:  vextracti64x4 $1, %zmm0, %ymm0
; CHECK-NEXT:  vcvtdq2pd
               %ymm0, %zmm0
; CHECK-NEXT:  vmovaps %zmm0, 64(%rdi)
; CHECK-NEXT:  vmovaps %zmm1, (%rdi)
; CHECK-NEXT:  vzeroupper
; CHECK-NEXT:  retq
               %mask = icmp slt <16 x i16> %a, zeroinitializer
               %1 = sitofp <16 x i1> %mask to <16 x double>
               store <16 x double> %1, <16 x double>* %res
               ret void
}

```

```

define dso_local void @ubto16f32_256(<16 x i16> %a, <16 x float>* %res) "min-legal-vector-width"="256" {
; CHECK-LABEL: ubto16f32_256:
; CHECK:      # %bb.0:
; CHECK-NEXT:  vpmovw2m %ymm0, %k0
; CHECK-NEXT:  kshiftrw $8, %k0, %k1
; CHECK-NEXT:  vpmovm2d %k1, %ymm0
; CHECK-NEXT:  vpsrld $31, %ymm0, %ymm0
; CHECK-NEXT:  vcvtdq2ps %ymm0, %ymm0
; CHECK-NEXT:  vpmovm2d %k0, %ymm1
; CHECK-NEXT:  vpsrld $31, %ymm1, %ymm1
; CHECK-NEXT:  vcvtdq2ps %ymm1, %ymm1
; CHECK-NEXT:  vmovaps %ymm1, (%rdi)
; CHECK-NEXT:  vmovaps %ymm0, 32(%rdi)
; CHECK-NEXT:  vzeroupper
; CHECK-NEXT:  retq
               %mask = icmp slt <16 x i16> %a, zeroinitializer
               %1 = uitofp <16 x i1> %mask to <16 x float>
               store <16 x float> %1, <16 x float>* %res
               ret void
}

```

```

define dso_local void @ubto16f32_512(<16 x i16> %a, <16 x float>* %res) "min-legal-vector-width"="512" {
; CHECK-LABEL: ubto16f32_512:
; CHECK:      # %bb.0:
; CHECK-NEXT:  vpmovw2m %ymm0, %k0
; CHECK-NEXT:  vpmovm2d %k0, %zmm0
; CHECK-NEXT:  vpsrld $31, %zmm0, %zmm0
; CHECK-NEXT:  vcvtdq2ps %zmm0, %zmm0

```

```

; CHECK-NEXT: vmovaps %zmm0, (%rdi)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%mask = icmp slt <16 x i16> %a, zeroinitializer
%1 = uitofp <16 x i1> %mask to <16 x float>
store <16 x float> %1, <16 x float>* %res
ret void
}

define dso_local void @ubto16f64_256(<16 x i16> %a, <16 x double>* %res) "min-legal-vector-width"="256" {
; CHECK-LABEL: ubto16f64_256:
; CHECK:      # %bb.0:
; CHECK-NEXT: vpmovw2m %ymm0, %k0
; CHECK-NEXT: kshiftrw $8, %k0, %k1
; CHECK-NEXT: vpmovm2d %k1, %ymm0
; CHECK-NEXT: vpsrld $31, %ymm0, %ymm0
; CHECK-NEXT: vcvtdq2pd %xmm0, %ymm1
; CHECK-NEXT: vextracti128
$1, %ymm0, %xmm0
; CHECK-NEXT: vcvtdq2pd %xmm0, %ymm0
; CHECK-NEXT: vpmovm2d %k0, %ymm2
; CHECK-NEXT: vpsrld $31, %ymm2, %ymm2
; CHECK-NEXT: vcvtdq2pd %xmm2, %ymm3
; CHECK-NEXT: vextracti128 $1, %ymm2, %xmm2
; CHECK-NEXT: vcvtdq2pd %xmm2, %ymm2
; CHECK-NEXT: vmovaps %ymm2, 32(%rdi)
; CHECK-NEXT: vmovaps %ymm3, (%rdi)
; CHECK-NEXT: vmovaps %ymm0, 96(%rdi)
; CHECK-NEXT: vmovaps %ymm1, 64(%rdi)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%mask = icmp slt <16 x i16> %a, zeroinitializer
%1 = uitofp <16 x i1> %mask to <16 x double>
store <16 x double> %1, <16 x double>* %res
ret void
}

define dso_local void @ubto16f64_512(<16 x i16> %a, <16 x double>* %res) "min-legal-vector-width"="512" {
; CHECK-LABEL: ubto16f64_512:
; CHECK:      # %bb.0:
; CHECK-NEXT: vpmovw2m %ymm0, %k0
; CHECK-NEXT: vpmovm2d %k0, %zmm0
; CHECK-NEXT: vpsrld $31, %zmm0, %zmm0
; CHECK-NEXT: vcvtdq2pd %ymm0, %zmm1
; CHECK-NEXT: vextracti64x4 $1, %zmm0, %ymm0
; CHECK-NEXT:
vcvtdq2pd %ymm0, %zmm0
; CHECK-NEXT: vmovaps %zmm0, 64(%rdi)

```

```

; CHECK-NEXT: vmovaps %zmm1, (%rdi)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%mask = icmp slt <16 x i16> %a, zeroinitializer
%1 = uitofp <16 x i1> %mask to <16 x double>
store <16 x double> %1, <16 x double>* %res
ret void
}

define <16 x i16> @test_16f32toub_256(<16 x float>* %ptr, <16 x i16> %passthru) "min-legal-vector-
width"="256" {
; CHECK-LABEL: test_16f32toub_256:
; CHECK:    # %bb.0:
; CHECK-NEXT: vcvttps2dq (%rdi), %ymm1
; CHECK-NEXT: vpslld $31, %ymm1, %ymm1
; CHECK-NEXT: vpmovd2m %ymm1, %k0
; CHECK-NEXT: vcvttps2dq 32(%rdi), %ymm1
; CHECK-NEXT: vpslld $31, %ymm1, %ymm1
; CHECK-NEXT: vpmovd2m %ymm1, %k1
; CHECK-NEXT: kunpckbw %k0, %k1, %k1
; CHECK-NEXT: vmovdqu16 %ymm0, %ymm0 {%k1} {z}
; CHECK-NEXT: retq
%a = load <16 x float>, <16 x float>* %ptr
%mask = fptoui <16 x float> %a to <16 x i1>
%select = select <16 x i1> %mask, <16 x i16> %passthru,
<16 x i16> zeroinitializer
ret <16 x i16> %select
}

define <16 x i16> @test_16f32toub_512(<16 x float>* %ptr, <16 x i16> %passthru) "min-legal-vector-
width"="512" {
; CHECK-LABEL: test_16f32toub_512:
; CHECK:    # %bb.0:
; CHECK-NEXT: vcvttps2dq (%rdi), %zmm1
; CHECK-NEXT: vpslld $31, %zmm1, %zmm1
; CHECK-NEXT: vpmovd2m %zmm1, %k1
; CHECK-NEXT: vmovdqu16 %ymm0, %ymm0 {%k1} {z}
; CHECK-NEXT: retq
%a = load <16 x float>, <16 x float>* %ptr
%mask = fptoui <16 x float> %a to <16 x i1>
%select = select <16 x i1> %mask, <16 x i16> %passthru, <16 x i16> zeroinitializer
ret <16 x i16> %select
}

define <16 x i16> @test_16f32tosb_256(<16 x float>* %ptr, <16 x i16> %passthru) "min-legal-vector-
width"="256" {
; CHECK-LABEL: test_16f32tosb_256:
; CHECK:    # %bb.0:

```

```
; CHECK-NEXT: vcvttps2dq (%rdi), %ymm1
; CHECK-NEXT: vpmovd2m %ymm1, %k0
; CHECK-NEXT: vcvttps2dq 32(%rdi), %ymm1
; CHECK-NEXT: vpmovd2m %ymm1, %k1
; CHECK-NEXT: kunpckbw %k0, %k1, %k1
; CHECK-NEXT:
    vmovdqu16 %ymm0, %ymm0 {%k1} {z}
; CHECK-NEXT: retq
%a = load <16 x float>, <16 x float>* %ptr
%mask = fptosi <16 x float> %a to <16 x i1>
%select = select <16 x i1> %mask, <16 x i16> %passthru, <16 x i16> zeroinitializer
ret <16 x i16> %select
}
```

```
define <16 x i16> @test_16f32tosb_512(<16 x float>* %ptr, <16 x i16> %passthru) "min-legal-vector-
width"="512" {
```

```
; CHECK-LABEL: test_16f32tosb_512:
; CHECK:      # %bb.0:
; CHECK-NEXT: vcvttps2dq (%rdi), %zmm1
; CHECK-NEXT: vpmovd2m %zmm1, %k1
; CHECK-NEXT: vmovdqu16 %ymm0, %ymm0 {%k1} {z}
; CHECK-NEXT: retq
%a = load <16 x float>, <16 x float>* %ptr
%mask = fptosi <16 x float> %a to <16 x i1>
%select = select <16 x i1> %mask, <16 x i16> %passthru, <16 x i16> zeroinitializer
ret <16 x i16> %select
}
```

```
define dso_local void @mul256(<64 x i8>* %a, <64 x i8>* %b, <64 x i8>* %c) "min-legal-vector-width"="256" {
```

```
; CHECK-SKX-VBMI-LABEL: mul256:
; CHECK-SKX-VBMI:      # %bb.0:
; CHECK-SKX-VBMI-NEXT: vmovdqa
    (%rdi), %ymm0
; CHECK-SKX-VBMI-NEXT: vmovdqa 32(%rdi), %ymm1
; CHECK-SKX-VBMI-NEXT: vmovdqa (%rsi), %ymm2
; CHECK-SKX-VBMI-NEXT: vmovdqa 32(%rsi), %ymm3
; CHECK-SKX-VBMI-NEXT: vpunpckhbw {{.*#+}} %ymm4 =
    ymm3[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-SKX-VBMI-NEXT: vpunpckhbw {{.*#+}} %ymm5 =
    ymm1[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-SKX-VBMI-NEXT: vpmullw %ymm4, %ymm5, %ymm4
; CHECK-SKX-VBMI-NEXT: vpunpcklbw {{.*#+}} %ymm3 =
    ymm3[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-SKX-VBMI-NEXT: vpunpcklbw {{.*#+}} %ymm1 =
    ymm1[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-SKX-VBMI-NEXT: vpmullw %ymm3, %ymm1, %ymm1
; CHECK-SKX-VBMI-NEXT: vmovdqa {{.*#+}} %ymm3 =
    [0,2,4,6,8,10,12,14,32,34,36,38,40,42,44,46,16,18,20,22,24,26,28,30,48,50,52,54,56,58,60,62]
```

```

;
CHECK-SKX-VBMI-NEXT: vpermt2b %ymm4, %ymm3, %ymm1
; CHECK-SKX-VBMI-NEXT: vpunpckhbw {{.*#+}} ymm4 =
ymm2[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-SKX-VBMI-NEXT: vpunpckhbw {{.*#+}} ymm5 =
ymm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-SKX-VBMI-NEXT: vpmullw %ymm4, %ymm5, %ymm4
; CHECK-SKX-VBMI-NEXT: vpunpcklbw {{.*#+}} ymm2 =
ymm2[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-SKX-VBMI-NEXT: vpunpcklbw {{.*#+}} ymm0 =
ymm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-SKX-VBMI-NEXT: vpmullw %ymm2, %ymm0, %ymm0
; CHECK-SKX-VBMI-NEXT: vpermt2b %ymm4, %ymm3, %ymm0
; CHECK-SKX-VBMI-NEXT: vmovdqa %ymm0, (%rdx)
; CHECK-SKX-VBMI-NEXT: vmovdqa %ymm1, 32(%rdx)
; CHECK-SKX-VBMI-NEXT: vzeroupper
; CHECK-SKX-VBMI-NEXT: retq
;
; CHECK-AVX512-LABEL: mul256:
; CHECK-AVX512:
# %bb.0:
; CHECK-AVX512-NEXT: vmovdqa (%rdi), %ymm0
; CHECK-AVX512-NEXT: vmovdqa 32(%rdi), %ymm1
; CHECK-AVX512-NEXT: vmovdqa (%rsi), %ymm2
; CHECK-AVX512-NEXT: vmovdqa 32(%rsi), %ymm3
; CHECK-AVX512-NEXT: vpunpckhbw {{.*#+}} ymm4 =
ymm3[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-AVX512-NEXT: vpunpckhbw {{.*#+}} ymm5 =
ymm1[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-AVX512-NEXT: vpmullw %ymm4, %ymm5, %ymm4
; CHECK-AVX512-NEXT: vpbroadcastw {{.*#+}} ymm5 =
[255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255]
; CHECK-AVX512-NEXT: vpand %ymm5, %ymm4, %ymm4
; CHECK-AVX512-NEXT: vpunpcklbw {{.*#+}} ymm3 =
ymm3[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-AVX512-NEXT: vpunpcklbw {{.*#+}} ymm1 =
ymm1[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
;
CHECK-AVX512-NEXT: vpmullw %ymm3, %ymm1, %ymm1
; CHECK-AVX512-NEXT: vpand %ymm5, %ymm1, %ymm1
; CHECK-AVX512-NEXT: vpackuswb %ymm4, %ymm1, %ymm1
; CHECK-AVX512-NEXT: vpunpckhbw {{.*#+}} ymm3 =
ymm2[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-AVX512-NEXT: vpunpckhbw {{.*#+}} ymm4 =
ymm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-AVX512-NEXT: vpmullw %ymm3, %ymm4, %ymm3
; CHECK-AVX512-NEXT: vpand %ymm5, %ymm3, %ymm3
; CHECK-AVX512-NEXT: vpunpcklbw {{.*#+}} ymm2 =

```

```

ymm2[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-AVX512-NEXT:  vpunpcklbw {{.*#+}} ymm0 =
ymm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-AVX512-NEXT:  vpmullw %ymm2, %ymm0, %ymm0
; CHECK-AVX512-NEXT:  vpand %ymm5, %ymm0, %ymm0
; CHECK-AVX512-NEXT:  vpackuswb %ymm3, %ymm0, %ymm0
; CHECK-AVX512-NEXT:
    vmovdqa %ymm0, (%rdx)
; CHECK-AVX512-NEXT:  vmovdqa %ymm1, 32(%rdx)
; CHECK-AVX512-NEXT:  vzeroupper
; CHECK-AVX512-NEXT:  retq
;
; CHECK-VBMI-LABEL: mul256:
; CHECK-VBMI:        # %bb.0:
; CHECK-VBMI-NEXT:  vmovdqa (%rdi), %ymm0
; CHECK-VBMI-NEXT:  vmovdqa 32(%rdi), %ymm1
; CHECK-VBMI-NEXT:  vmovdqa (%rsi), %ymm2
; CHECK-VBMI-NEXT:  vmovdqa 32(%rsi), %ymm3
; CHECK-VBMI-NEXT:  vpunpckhbw {{.*#+}} ymm4 =
ymm3[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-VBMI-NEXT:  vpunpckhbw {{.*#+}} ymm5 =
ymm1[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-VBMI-NEXT:  vpmullw %ymm4, %ymm5, %ymm4
; CHECK-VBMI-NEXT:  vpunpcklbw {{.*#+}} ymm3 =
ymm3[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-VBMI-NEXT:  vpunpcklbw {{.*#+}} ymm1 =
ymm1[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
;
    CHECK-VBMI-NEXT:  vpmullw %ymm3, %ymm1, %ymm1
; CHECK-VBMI-NEXT:  vmovdqa {{.*#+}} ymm3 =
[0,2,4,6,8,10,12,14,32,34,36,38,40,42,44,46,16,18,20,22,24,26,28,30,48,50,52,54,56,58,60,62]
; CHECK-VBMI-NEXT:  vpermt2b %ymm4, %ymm3, %ymm1
; CHECK-VBMI-NEXT:  vpunpckhbw {{.*#+}} ymm4 =
ymm2[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-VBMI-NEXT:  vpunpckhbw {{.*#+}} ymm5 =
ymm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-VBMI-NEXT:  vpmullw %ymm4, %ymm5, %ymm4
; CHECK-VBMI-NEXT:  vpunpcklbw {{.*#+}} ymm2 =
ymm2[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-VBMI-NEXT:  vpunpcklbw {{.*#+}} ymm0 =
ymm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-VBMI-NEXT:  vpmullw %ymm2, %ymm0, %ymm0
; CHECK-VBMI-NEXT:  vpermt2b %ymm4, %ymm3, %ymm0
; CHECK-VBMI-NEXT:  vmovdqa %ymm0, (%rdx)
;
    CHECK-VBMI-NEXT:  vmovdqa %ymm1, 32(%rdx)
; CHECK-VBMI-NEXT:  vzeroupper
; CHECK-VBMI-NEXT:  retq

```

```

%d = load <64 x i8>, <64 x i8>* %a
%e = load <64 x i8>, <64 x i8>* %b
%f = mul <64 x i8> %d, %e
store <64 x i8> %f, <64 x i8>* %c
ret void
}

```

```

define dso_local void @mul512(<64 x i8>* %a, <64 x i8>* %b, <64 x i8>* %c) "min-legal-vector-width"="512" {
; CHECK-SKX-VBMI-LABEL: mul512:
; CHECK-SKX-VBMI:      # %bb.0:
; CHECK-SKX-VBMI-NEXT:  vmovdqa64 (%rdi), %zmm0
; CHECK-SKX-VBMI-NEXT:  vmovdqa64 (%rsi), %zmm1
; CHECK-SKX-VBMI-NEXT:  vpunpckhbw {{.*#+}} zmm2 =
zmm1[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31,40,40,41,41,4
2,42,43,43,44,44,45,45,46,46,47,47,56,56,57,57,58,58,59,59,60,60,61,61,62,62,63,63]
; CHECK-SKX-VBMI-NEXT:  vpunpckhbw {{.*#+}} zmm3 =
zmm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31,40,40,41,41,4
2,42,43,43,44,44,45,45,46,46,47,47,56,56,57,57,58,58,59,59,60,60,61,61,62,62,63,63]
;
; CHECK-SKX-VBMI-NEXT:  vpmullw %zmm2, %zmm3, %zmm2
; CHECK-SKX-VBMI-NEXT:  vpunpcklbw {{.*#+}} zmm1 =
zmm1[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23,32,32,33,33,34,34,35,35,36,3
6,37,37,38,38,39,39,48,48,49,49,50,50,51,51,52,52,53,53,54,54,55,55]
; CHECK-SKX-VBMI-NEXT:  vpunpcklbw {{.*#+}} zmm0 =
zmm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23,32,32,33,33,34,34,35,35,36,3
6,37,37,38,38,39,39,48,48,49,49,50,50,51,51,52,52,53,53,54,54,55,55]
; CHECK-SKX-VBMI-NEXT:  vpmullw %zmm1, %zmm0, %zmm0
; CHECK-SKX-VBMI-NEXT:  vmovdqa64 {{.*#+}} zmm1 =
[0,2,4,6,8,10,12,14,64,66,68,70,72,74,76,78,16,18,20,22,24,26,28,30,80,82,84,86,88,90,92,94,32,34,36,38,40,42,44,
46,96,98,100,102,104,106,108,110,48,50,52,54,56,58,60,62,112,114,116,118,120,122,124,126]
; CHECK-SKX-VBMI-NEXT:  vpermi2b %zmm2, %zmm0, %zmm1
; CHECK-SKX-VBMI-NEXT:  vmovdqa64 %zmm1, (%rdx)
; CHECK-SKX-VBMI-NEXT:  vzeroupper
;
; CHECK-SKX-VBMI-NEXT:  retq
;
; CHECK-AVX512-LABEL: mul512:
; CHECK-AVX512:      # %bb.0:
; CHECK-AVX512-NEXT:  vmovdqa64 (%rdi), %zmm0
; CHECK-AVX512-NEXT:  vmovdqa64 (%rsi), %zmm1
; CHECK-AVX512-NEXT:  vpunpckhbw {{.*#+}} zmm2 =
zmm1[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31,40,40,41,41,4
2,42,43,43,44,44,45,45,46,46,47,47,56,56,57,57,58,58,59,59,60,60,61,61,62,62,63,63]
; CHECK-AVX512-NEXT:  vpunpckhbw {{.*#+}} zmm3 =
zmm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31,40,40,41,41,4
2,42,43,43,44,44,45,45,46,46,47,47,56,56,57,57,58,58,59,59,60,60,61,61,62,62,63,63]
; CHECK-AVX512-NEXT:  vpmullw %zmm2, %zmm3, %zmm2
; CHECK-AVX512-NEXT:  vpbroadcastw {{.*#+}} zmm3 =

```

```
[255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255]
; CHECK-AVX512-NEXT:    vpandq %zmm3, %zmm2, %zmm2
; CHECK-AVX512-NEXT:    vpunpcklbw
{{.*#+}} zmm1 =
zmm1[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23,32,32,33,33,34,34,35,35,36,3
6,37,37,38,38,39,39,48,48,49,49,50,50,51,51,52,52,53,53,54,54,55,55]
; CHECK-AVX512-NEXT:    vpunpcklbw {{.*#+}} zmm0 =
zmm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23,32,32,33,33,34,34,35,35,36,3
6,37,37,38,38,39,39,48,48,49,49,50,50,51,51,52,52,53,53,54,54,55,55]
; CHECK-AVX512-NEXT:    vpmullw %zmm1, %zmm0, %zmm0
; CHECK-AVX512-NEXT:    vpandq %zmm3, %zmm0, %zmm0
; CHECK-AVX512-NEXT:    vpackuswb %zmm2, %zmm0, %zmm0
; CHECK-AVX512-NEXT:    vmovdqa64 %zmm0, (%rdx)
; CHECK-AVX512-NEXT:    vzeroupper
; CHECK-AVX512-NEXT:    retq
;
; CHECK-VBMI-LABEL: mul512:
; CHECK-VBMI:          # %bb.0:
; CHECK-VBMI-NEXT:     vmovdqa64 (%rdi), %zmm0
; CHECK-VBMI-NEXT:     vmovdqa64 (%rsi), %zmm1
; CHECK-VBMI-NEXT:     vpunpckhbw {{.*#+}} zmm2 =
zmm1[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31,40,40,41,41,4
2,42,43,43,44,44,45,45,46,46,47,47,56,56,57,57,58,58,59,59,60,60,61,61,62,62,63,63]
;
CHECK-VBMI-NEXT:      vpunpckhbw {{.*#+}} zmm3 =
zmm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31,40,40,41,41,4
2,42,43,43,44,44,45,45,46,46,47,47,56,56,57,57,58,58,59,59,60,60,61,61,62,62,63,63]
; CHECK-VBMI-NEXT:     vpmullw %zmm2, %zmm3, %zmm2
; CHECK-VBMI-NEXT:     vpunpcklbw {{.*#+}} zmm1 =
zmm1[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23,32,32,33,33,34,34,35,35,36,3
6,37,37,38,38,39,39,48,48,49,49,50,50,51,51,52,52,53,53,54,54,55,55]
; CHECK-VBMI-NEXT:     vpunpcklbw {{.*#+}} zmm0 =
zmm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23,32,32,33,33,34,34,35,35,36,3
6,37,37,38,38,39,39,48,48,49,49,50,50,51,51,52,52,53,53,54,54,55,55]
; CHECK-VBMI-NEXT:     vpmullw %zmm1, %zmm0, %zmm0
; CHECK-VBMI-NEXT:     vmovdqa64 {{.*#+}} zmm1 =
[0,2,4,6,8,10,12,14,64,66,68,70,72,74,76,78,16,18,20,22,24,26,28,30,80,82,84,86,88,90,92,94,32,34,36,38,40,42,44,
46,96,98,100,102,104,106,108,110,48,50,52,54,56,58,60,62,112,114,116,118,120,122,124,126]
;
CHECK-VBMI-NEXT:      vpermi2b %zmm2, %zmm0, %zmm1
; CHECK-VBMI-NEXT:     vmovdqa64 %zmm1, (%rdx)
; CHECK-VBMI-NEXT:     vzeroupper
; CHECK-VBMI-NEXT:     retq
%d = load <64 x i8>, <64 x i8>* %a
%e = load <64 x i8>, <64 x i8>* %b
%f = mul <64 x i8>%d,%e
store <64 x i8>%f,<64 x i8>* %c
```



```
ret void
}
```

; This threw an assertion at one point.

```
define <4 x i32> @mload_v4i32(<4 x i32> %trigger, <4 x i32>* %addr, <4 x i32> %dst) "min-legal-vector-
width"="256" {
; CHECK-LABEL: mload_v4i32:
; CHECK:      # %bb.0:
; CHECK-NEXT: vptestnmd %xmm0, %xmm0, %k1
; CHECK-NEXT: vpblendmd (%rdi), %xmm1, %xmm0 {%k1}
; CHECK-NEXT: retq
%mask = icmp eq <4 x i32> %trigger, zeroinitializer
%res = call <4 x i32> @llvm.masked.load.v4i32.p0v4i32(<4 x i32>* %addr, i32 4, <4 x i1> %mask, <4 x i32>
%dst)
ret <4 x i32> %res
}
declare <4 x i32>
@llvm.masked.load.v4i32.p0v4i32(<4 x i32>*, i32, <4 x i1>, <4 x i32>)
```

```
define <16 x i32> @trunc_v16i64_v16i32(<16 x i64>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_v16i64_v16i32:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rdi), %ymm0
; CHECK-NEXT: vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT: vmovdqa 64(%rdi), %ymm2
; CHECK-NEXT: vmovdqa 96(%rdi), %ymm3
; CHECK-NEXT: vpmovqd %ymm0, %xmm0
; CHECK-NEXT: vpmovqd %ymm1, %xmm1
; CHECK-NEXT: vinserti128 $1, %xmm1, %ymm0, %ymm0
; CHECK-NEXT: vpmovqd %ymm2, %xmm1
; CHECK-NEXT: vpmovqd %ymm3, %xmm2
; CHECK-NEXT: vinserti128 $1, %xmm2, %ymm1, %ymm1
; CHECK-NEXT: retq
%a = load <16 x i64>, <16 x i64>* %x
%b = trunc <16 x i64> %a to <16 x i32>
ret <16 x i32> %b
}
```

```
define <16 x i8> @trunc_v16i64_v16i8(<16 x i64>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_v16i64_v16i8:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rdi), %ymm0
; CHECK-NEXT: vmovdqa 32(%rdi),
%ymm1
; CHECK-NEXT: vmovdqa 64(%rdi), %ymm2
; CHECK-NEXT: vmovdqa 96(%rdi), %ymm3
; CHECK-NEXT: vpmovqb %ymm3, %xmm3
; CHECK-NEXT: vpmovqb %ymm2, %xmm2
```

```

; CHECK-NEXT: vpunpckldq {{.*#+}} xmm2 = xmm2[0],xmm3[0],xmm2[1],xmm3[1]
; CHECK-NEXT: vpmovqb %ymm1, %xmm1
; CHECK-NEXT: vpmovqb %ymm0, %xmm0
; CHECK-NEXT: vpunpckldq {{.*#+}} xmm0 = xmm0[0],xmm1[0],xmm0[1],xmm1[1]
; CHECK-NEXT: vpunpckldq {{.*#+}} xmm0 = xmm0[0],xmm2[0]
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%a = load <16 x i64>, <16 x i64>* %x
%b = trunc <16 x i64> %a to <16 x i8>
ret <16 x i8> %b
}

```

```

define <16 x i8> @trunc_v16i32_v16i8(<16 x i32>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_v16i32_v16i8:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rdi), %ymm0
; CHECK-NEXT: vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT: vpmovdb %ymm1, %xmm1
; CHECK-NEXT: vpmovdb %ymm0, %xmm0
; CHECK-NEXT: vpunpckldq {{.*#+}} xmm0 = xmm0[0],xmm1[0]
; CHECK-NEXT: vzeroupper
;
CHECK-NEXT: retq
%a = load <16 x i32>, <16 x i32>* %x
%b = trunc <16 x i32> %a to <16 x i8>
ret <16 x i8> %b
}

```

```

define <8 x i8> @trunc_v8i64_v8i8(<8 x i64>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_v8i64_v8i8:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rdi), %ymm0
; CHECK-NEXT: vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT: vpmovqb %ymm1, %xmm1
; CHECK-NEXT: vpmovqb %ymm0, %xmm0
; CHECK-NEXT: vpunpckldq {{.*#+}} xmm0 = xmm0[0],xmm1[0],xmm0[1],xmm1[1]
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%a = load <8 x i64>, <8 x i64>* %x
%b = trunc <8 x i64> %a to <8 x i8>
ret <8 x i8> %b
}

```

```

define <8 x i16> @trunc_v8i64_v8i16(<8 x i64>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_v8i64_v8i16:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rdi), %ymm0
; CHECK-NEXT: vmovdqa 32(%rdi), %ymm1

```

```

; CHECK-NEXT: vpmovqw %ymm1, %xmm1
; CHECK-NEXT: vpmovqw %ymm0, %xmm0
; CHECK-NEXT: vpunpcklqdq {{.*#+}} xmm0
= xmm0[0],xmm1[0]
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%a = load <8 x i64>, <8 x i64>* %x
%b = trunc <8 x i64> %a to <8 x i16>
ret <8 x i16> %b
}

define <8 x i32> @trunc_v8i64_v8i32_zeroes(<8 x i64>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_v8i64_v8i32_zeroes:
; CHECK:      # %bb.0:
; CHECK-NEXT: vpsrlq $48, 32(%rdi), %ymm0
; CHECK-NEXT: vpsrlq $48, (%rdi), %ymm1
; CHECK-NEXT: vpackusdw %ymm0, %ymm1, %ymm0
; CHECK-NEXT: vpermq {{.*#+}} ymm0 = ymm0[0,2,1,3]
; CHECK-NEXT: retq
%a = load <8 x i64>, <8 x i64>* %x
%b = lshr <8 x i64> %a, <i64 48, i64 48, i64 48, i64 48, i64 48, i64 48, i64 48, i64 48>
%c = trunc <8 x i64> %b to <8 x i32>
ret <8 x i32> %c
}

define <16 x i16> @trunc_v16i32_v16i16_zeroes(<16 x i32>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_v16i32_v16i16_zeroes:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rdi), %ymm1
; CHECK-NEXT: vmovdqa {{.*#+}} ymm0 = [1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31]
;
CHECK-NEXT: vpermi2w 32(%rdi), %ymm1, %ymm0
; CHECK-NEXT: retq
%a = load <16 x i32>, <16 x i32>* %x
%b = lshr <16 x i32> %a, <i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16>
%c = trunc <16 x i32> %b to <16 x i16>
ret <16 x i16> %c
}

define <32 x i8> @trunc_v32i16_v32i8_zeroes(<32 x i16>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-SKX-VBMI-LABEL: trunc_v32i16_v32i8_zeroes:
; CHECK-SKX-VBMI:      # %bb.0:
; CHECK-SKX-VBMI-NEXT: vmovdqa (%rdi), %ymm1
; CHECK-SKX-VBMI-NEXT: vmovdqa {{.*#+}} ymm0 =
[1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35,37,39,41,43,45,47,49,51,53,55,57,59,61,63]
; CHECK-SKX-VBMI-NEXT: vpermi2b 32(%rdi), %ymm1, %ymm0
; CHECK-SKX-VBMI-NEXT: retq

```

```

;
; CHECK-AVX512-LABEL: trunc_v32i16_v32i8_zeroes:
; CHECK-AVX512:    # %bb.0:
; CHECK-AVX512-NEXT:  vpsrlw $8, 32(%rdi), %ymm0
; CHECK-AVX512-NEXT:  vpsrlw $8, (%rdi),
    %ymm1
; CHECK-AVX512-NEXT:  vpackuswb %ymm0, %ymm1, %ymm0
; CHECK-AVX512-NEXT:  vpermq {{.*#+}} ymm0 = ymm0[0,2,1,3]
; CHECK-AVX512-NEXT:  retq
;
; CHECK-VBMI-LABEL: trunc_v32i16_v32i8_zeroes:
; CHECK-VBMI:    # %bb.0:
; CHECK-VBMI-NEXT:  vmovdqa (%rdi), %ymm1
; CHECK-VBMI-NEXT:  vmovdqa {{.*#+}} ymm0 =
[1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35,37,39,41,43,45,47,49,51,53,55,57,59,61,63]
; CHECK-VBMI-NEXT:  vperm2b 32(%rdi), %ymm1, %ymm0
; CHECK-VBMI-NEXT:  retq
    %a = load <32 x i16>, <32 x i16>* %x
    %b = lshr <32 x i16> %a, <i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8>
    %c = trunc <32 x i16> %b to <32 x i8>
    ret <32 x i8> %c
}

```

```

define <8 x i32> @trunc_v8i64_v8i32_sign(<8 x i64>* %x) nounwind "min-legal-vector-width"="256" {

```

```

; CHECK-LABEL: trunc_v8i64_v8i32_sign:
; CHECK:
    # %bb.0:
; CHECK-NEXT:  vpsraq $48, 32(%rdi), %ymm0
; CHECK-NEXT:  vpsraq $48, (%rdi), %ymm1
; CHECK-NEXT:  vpackssdw %ymm0, %ymm1, %ymm0
; CHECK-NEXT:  vpermq {{.*#+}} ymm0 = ymm0[0,2,1,3]
; CHECK-NEXT:  retq
    %a = load <8 x i64>, <8 x i64>* %x
    %b = ashr <8 x i64> %a, <i64 48, i64 48, i64 48, i64 48, i64 48, i64 48, i64 48, i64 48>
    %c = trunc <8 x i64> %b to <8 x i32>
    ret <8 x i32> %c
}

```

```

define <16 x i16> @trunc_v16i32_v16i16_sign(<16 x i32>* %x) nounwind "min-legal-vector-width"="256" {

```

```

; CHECK-LABEL: trunc_v16i32_v16i16_sign:
; CHECK:    # %bb.0:
; CHECK-NEXT:  vmovdqa (%rdi), %ymm1
; CHECK-NEXT:  vmovdqa {{.*#+}} ymm0 = [1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31]
; CHECK-NEXT:  vperm2w 32(%rdi), %ymm1, %ymm0
; CHECK-NEXT:  retq
    %a = load <16 x i32>, <16 x i32>* %x
    %b = ashr <16 x i32> %a, <i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16>

```

```
16, i32 16, i32 16, i32 16, i32 16>
%c = trunc <16 x i32> %b to <16
x i16>
ret <16 x i16> %c
}
```

```
define dso_local void @zext_v16i8_v16i64(<16 x i8> %x, <16 x i64>* %y) nounwind "min-legal-vector-
width"="256" {
; CHECK-LABEL: zext_v16i8_v16i64:
; CHECK:      # %bb.0:
; CHECK-NEXT:  vpmovzxbw {{.*#+}} ymm1 =
xmm0[0],zero,xmm0[1],zero,xmm0[2],zero,xmm0[3],zero,xmm0[4],zero,xmm0[5],zero,xmm0[6],zero,xmm0[7],zer
o,xmm0[8],zero,xmm0[9],zero,xmm0[10],zero,xmm0[11],zero,xmm0[12],zero,xmm0[13],zero,xmm0[14],zero,xm
m0[15],zero
; CHECK-NEXT:  vpshufd {{.*#+}} xmm2 = xmm1[2,3,2,3]
; CHECK-NEXT:
```

```

    vpmovzxwq {{.*#+}} ymm2 =
xmm2[0],zero,zero,zero,xmm2[1],zero,zero,zero,xmm2[2],zero,zero,zero,xmm2[3],zero,zero,zero
; CHECK-NEXT:    vextracti128 $1, %ymm1, %xmm1
; CHECK-NEXT:    vpshufd {{.*#+}} xmm3 = xmm1[2,3,2,3]
; CHECK-NEXT:    vpmovzxwq {{.*#+}} ymm3 =
xmm3[0],zero,zero,zero,xmm3[1],zero,zero,zero,xmm3[2],zero,zero,zero,xmm3[3],zero,zero,zero
; CHECK-NEXT:    vpmovzxwq {{.*#+}} ymm1 =
xmm1[0],zero,zero,zero,xmm1[1],zero,zero,zero,xmm1[2],zero,zero,zero,xmm1[3],zero,zero,zero
; CHECK-NEXT:    vpmovzxbq {{.*#+}} ymm0 =
xmm0[0],zero,zero,zero,zero,zero,zero,zero,xmm0[1],zero,zero,zero,zero,zero,zero,zero,xmm0[2],zero,zero,zero,zero,zero,zero,zero,xmm0[3],zero,zero,zero,zero,zero,zero,zero
; CHECK-NEXT:    vmovdqa %ymm0, (%rdi)
; CHECK-NEXT:    vmovdqa %ymm1, 64(%rdi)
; CHECK-NEXT:    vmovdqa %ymm3, 96(%rdi)
; CHECK-NEXT:    vmovdqa %ymm2, 32(%rdi)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
    %a = zext <16 x i8> %x to <16 x i64>
    store <16 x i64> %a, <16 x i64>*
    %y
    ret void
}

```

```

define dso_local void @sext_v16i8_v16i64(<16 x i8> %x, <16 x i64>* %y) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: sext_v16i8_v16i64:
; CHECK:      # %bb.0:
; CHECK-NEXT: vpmovsxbw %xmm0, %ymm1
; CHECK-NEXT: vpshufd {{.*#+}} xmm2 = xmm1[2,3,2,3]
; CHECK-NEXT: vpmovswwq %xmm2, %ymm2
; CHECK-NEXT: vextracti128 $1, %ymm1, %xmm1
; CHECK-NEXT: vpshufd {{.*#+}} xmm3 = xmm1[2,3,2,3]
; CHECK-NEXT: vpmovswwq %xmm3, %ymm3
; CHECK-NEXT: vpmovswwq %xmm1, %ymm1
; CHECK-NEXT: vpmovsxbq %xmm0, %ymm0
; CHECK-NEXT: vmovdqa %ymm0, (%rdi)
; CHECK-NEXT: vmovdqa %ymm1, 64(%rdi)
; CHECK-NEXT: vmovdqa %ymm3, 96(%rdi)
; CHECK-NEXT: vmovdqa %ymm2, 32(%rdi)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
    %a = sext <16 x i8> %x to <16 x i64>
    store <16 x i64> %a, <16 x i64>* %y
    ret void
}

```

```

define dso_local void @vselect_split_v8i16_setcc(<8 x i16> %s, <8 x i16> %t, <8 x i64>* %p, <8 x i64>* %q, <8 x i64>* %r) "min-legal-vector-width"="256"

```

```

{
; CHECK-LABEL: vselect_split_v8i16_setcc:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rsi), %ymm2
; CHECK-NEXT: vmovdqa 32(%rsi), %ymm3
; CHECK-NEXT: vpcmpeqw %xmm1, %xmm0, %k1
; CHECK-NEXT: kshiftrb $4, %k1, %k2
; CHECK-NEXT: vmovdqa64 32(%rdi), %ymm3 {%k2}
; CHECK-NEXT: vmovdqa64 (%rdi), %ymm2 {%k1}
; CHECK-NEXT: vmovdqa %ymm2, (%rdx)
; CHECK-NEXT: vmovdqa %ymm3, 32(%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%x = load <8 x i64>, <8 x i64>* %p
%y = load <8 x i64>, <8 x i64>* %q
%a = icmp eq <8 x i16> %s, %t
%b = select <8 x i1> %a, <8 x i64> %x, <8 x i64> %y
store <8 x i64> %b, <8 x i64>* %r
ret void
}

```

```

define dso_local void @vselect_split_v8i32_setcc(<8 x i32> %s, <8 x i32> %t, <8 x i64>* %p, <8 x i64>* %q, <8
x i64>* %r) "min-legal-vector-width"="256" {
; CHECK-LABEL: vselect_split_v8i32_setcc:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rsi), %ymm2
; CHECK-NEXT: vmovdqa 32(%rsi), %ymm3
; CHECK-NEXT: vpcmpeqd %ymm1,
%ymm0, %k1
; CHECK-NEXT: kshiftrb $4, %k1, %k2
; CHECK-NEXT: vmovdqa64 32(%rdi), %ymm3 {%k2}
; CHECK-NEXT: vmovdqa64 (%rdi), %ymm2 {%k1}
; CHECK-NEXT: vmovdqa %ymm2, (%rdx)
; CHECK-NEXT: vmovdqa %ymm3, 32(%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%x = load <8 x i64>, <8 x i64>* %p
%y = load <8 x i64>, <8 x i64>* %q
%a = icmp eq <8 x i32> %s, %t
%b = select <8 x i1> %a, <8 x i64> %x, <8 x i64> %y
store <8 x i64> %b, <8 x i64>* %r
ret void
}

```

```

define dso_local void @vselect_split_v16i8_setcc(<16 x i8> %s, <16 x i8> %t, <16 x i32>* %p, <16 x i32>* %q,
<16 x i32>* %r) "min-legal-vector-width"="256" {
; CHECK-LABEL: vselect_split_v16i8_setcc:
; CHECK:      # %bb.0:

```

```

; CHECK-NEXT: vmovdqa (%rsi), %ymm2
; CHECK-NEXT: vmovdqa 32(%rsi), %ymm3
; CHECK-NEXT: vpcmpeqb %xmm1, %xmm0, %k1
; CHECK-NEXT: kshiftrw $8, %k1, %k2
; CHECK-NEXT: vmovdqa32 32(%rdi), %ymm3 {%k2}
; CHECK-NEXT: vmovdqa32 (%rdi), %ymm2 {%k1}
; CHECK-NEXT: vmovdqa %ymm2,
(%rdx)
; CHECK-NEXT: vmovdqa %ymm3, 32(%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%x = load <16 x i32>, <16 x i32>* %p
%y = load <16 x i32>, <16 x i32>* %q
%a = icmp eq <16 x i8> %s, %t
%b = select <16 x i1> %a, <16 x i32> %x, <16 x i32> %y
store <16 x i32> %b, <16 x i32>* %r
ret void
}

```

```

define dso_local void @vselect_split_v16i16_setcc(<16 x i16> %s, <16 x i16> %t, <16 x i32>* %p, <16 x i32>*
%q, <16 x i32>* %r) "min-legal-vector-width"="256" {
; CHECK-LABEL: vselect_split_v16i16_setcc:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rsi), %ymm2
; CHECK-NEXT: vmovdqa 32(%rsi), %ymm3
; CHECK-NEXT: vpcmpeqw %ymm1, %ymm0, %k1
; CHECK-NEXT: kshiftrw $8, %k1, %k2
; CHECK-NEXT: vmovdqa32 32(%rdi), %ymm3 {%k2}
; CHECK-NEXT: vmovdqa32 (%rdi), %ymm2 {%k1}
; CHECK-NEXT: vmovdqa %ymm2, (%rdx)
; CHECK-NEXT: vmovdqa %ymm3, 32(%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%x = load <16 x i32>, <16 x i32>* %p
%y = load <16 x i32>,
<16 x i32>* %q
%a = icmp eq <16 x i16> %s, %t
%b = select <16 x i1> %a, <16 x i32> %x, <16 x i32> %y
store <16 x i32> %b, <16 x i32>* %r
ret void
}

```

```

define <16 x i8> @trunc_packus_v16i32_v16i8(<16 x i32>* %p) "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_packus_v16i32_v16i8:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rdi), %ymm0
; CHECK-NEXT: vpackusdw 32(%rdi), %ymm0, %ymm0
; CHECK-NEXT: vpermq { {.*,#+} } ymm0 = ymm0[0,2,1,3]

```



```

; CHECK-NEXT: vpmovuswb %ymm0, %xmm0
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%a = load <16 x i32>, <16 x i32>* %p
%b = icmp slt <16 x i32> %a, <i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255>
%c = select <16 x i1> %b, <16 x i32> %a, <16 x i32> <i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255>
%d = icmp sgt <16 x i32> %c, zeroinitializer
%e = select <16 x i1> %d, <16 x i32> %c, <16 x i32> zeroinitializer
%f = trunc <16 x i32> %e to <16 x i8>
ret <16 x i8> %f
}

```

```

define dso_local void @trunc_packus_v16i32_v16i8_store(<16 x i32>* %p, <16 x i8>* %q) "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_packus_v16i32_v16i8_store:
; CHECK:      # %bb.0:
; CHECK-NEXT: vmovdqa (%rdi), %ymm0
; CHECK-NEXT: vpackusdw 32(%rdi), %ymm0, %ymm0
; CHECK-NEXT: vpermq { {. *#+} } ymm0 = ymm0[0,2,1,3]
; CHECK-NEXT: vpmovuswb %ymm0, (%rsi)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%a = load <16 x i32>, <16 x i32>* %p
%b = icmp slt <16 x i32> %a, <i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255>
%c = select <16 x i1> %b, <16 x i32> %a, <16 x i32> <i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255>
%d = icmp sgt <16
x i32> %c, zeroinitializer
%e = select <16 x i1> %d, <16 x i32> %c, <16 x i32> zeroinitializer
%f = trunc <16 x i32> %e to <16 x i8>
store <16 x i8> %f, <16 x i8>* %q
ret void
}

```

```

define <64 x i1> @v64i1_argument_return(<64 x i1> %x) "min-legal-vector-width"="256" {
; CHECK-LABEL: v64i1_argument_return:
; CHECK:      # %bb.0:
; CHECK-NEXT: retq
ret <64 x i1> %x
}

```

```

define dso_local void @v64i1_shuffle(<64 x i8>* %x, <64 x i8>* %y) "min-legal-vector-width"="256" {
; CHECK-LABEL: v64i1_shuffle:
; CHECK:      # %bb.0: # %entry
; CHECK-NEXT: vmovdqa (%rdi), %ymm1

```

```

; CHECK-NEXT: vmovdqa 32(%rdi), %ymm0
; CHECK-NEXT: vptestnmb %ymm1, %ymm1, %k0
; CHECK-NEXT: kshiftrd $1, %k0, %k1
; CHECK-NEXT: kshiftrlq $63, %k0, %k2
; CHECK-NEXT: kshiftrq $62, %k2, %k2
; CHECK-NEXT: kshiftrlq $63, %k1, %k1
; CHECK-NEXT: kshiftrq $63, %k1, %k1
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-5, %rax
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1,
%k1
; CHECK-NEXT: kshiftrd $3, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $61, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-9, %rax
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $2, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $60, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-17, %rax
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $5, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $59, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-33, %rax
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $4, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $58, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-65, %rax
; CHECK-NEXT: kmovq %rax, %k2
;
CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $7, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $57, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-129, %rax
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1

```

```

; CHECK-NEXT: kshiftrd $6, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $56, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-257, %rax # imm = 0xFEFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $9, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $55, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-513, %rax # imm = 0xFDFD
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $8, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $54, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
;
CHECK-NEXT: movq $-1025, %rax # imm = 0xFBFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $11, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $53, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-2049, %rax # imm = 0xF7FF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $10, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $52, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-4097, %rax # imm = 0xEFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $13, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $51, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-8193, %rax # imm = 0xDFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $12, %k0, %k2
; CHECK-NEXT:
    kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $50, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1

```

```

; CHECK-NEXT:  movq $-16385, %rax # imm = 0xBFFF
; CHECK-NEXT:  kmovq %rax, %k2
; CHECK-NEXT:  kandq %k2, %k1, %k1
; CHECK-NEXT:  kshiftrd $15, %k0, %k2
; CHECK-NEXT:  kshiftrlq $63, %k2, %k2
; CHECK-NEXT:  kshiftrq $49, %k2, %k2
; CHECK-NEXT:  korq %k2, %k1, %k1
; CHECK-NEXT:  movq $-32769, %rax # imm = 0xFFFF7FFF
; CHECK-NEXT:  kmovq %rax, %k2
; CHECK-NEXT:  kandq %k2, %k1, %k1
; CHECK-NEXT:  kshiftrd $14, %k0, %k2
; CHECK-NEXT:  kshiftrlq $63, %k2, %k2
; CHECK-NEXT:  kshiftrq $48, %k2, %k2
; CHECK-NEXT:  korq %k2, %k1, %k1
; CHECK-NEXT:  movq $-65537, %rax # imm = 0xFFFFFFFFFF
; CHECK-NEXT:  kmovq %rax, %k2
; CHECK-NEXT:  kandq %k2, %k1, %k1
; CHECK-NEXT:  kshiftrd $17, %k0, %k2
; CHECK-NEXT:  kshiftrlq $63, %k2, %k2
; CHECK-NEXT:  kshiftrq $47, %k2, %k2
; CHECK-NEXT:  korq %k2, %k1, %k1
; CHECK-NEXT:  movq $-131073, %rax # imm = 0xFFFFDFFF
;
CHECK-NEXT:  kmovq %rax, %k2
; CHECK-NEXT:  kandq %k2, %k1, %k1
; CHECK-NEXT:  kshiftrd $16, %k0, %k2
; CHECK-NEXT:  kshiftrlq $63, %k2, %k2
; CHECK-NEXT:  kshiftrq $46, %k2, %k2
; CHECK-NEXT:  korq %k2, %k1, %k1
; CHECK-NEXT:  movq $-262145, %rax # imm = 0xFFFFBFFF
; CHECK-NEXT:  kmovq %rax, %k2
; CHECK-NEXT:  kandq %k2, %k1, %k1
; CHECK-NEXT:  kshiftrd $19, %k0, %k2
; CHECK-NEXT:  kshiftrlq $63, %k2, %k2
; CHECK-NEXT:  kshiftrq $45, %k2, %k2
; CHECK-NEXT:  korq %k2, %k1, %k1
; CHECK-NEXT:  movq $-524289, %rax # imm = 0xFFF7FFFF
; CHECK-NEXT:  kmovq %rax, %k2
; CHECK-NEXT:  kandq %k2, %k1, %k1
; CHECK-NEXT:  kshiftrd $18, %k0, %k2
; CHECK-NEXT:  kshiftrlq $63, %k2, %k2
; CHECK-NEXT:  kshiftrq $44, %k2, %k2
; CHECK-NEXT:  korq %k2, %k1, %k1
; CHECK-NEXT:  movq $-1048577, %rax # imm = 0xFFEFFFFF
; CHECK-NEXT:  kmovq %rax, %k2
; CHECK-NEXT:  kandq %k2, %k1, %k1
; CHECK-NEXT:  kshiftrd $21, %k0, %k2
; CHECK-NEXT:  kshiftrlq $63, %k2, %k2

```

```

; CHECK-NEXT:
    kshiftrq $43, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-2097153, %rax # imm = 0xFFDFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $20, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $42, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-4194305, %rax # imm = 0xFFBFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $23, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $41, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-8388609, %rax # imm = 0xFF7FFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $22, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $40, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-16777217, %rax # imm = 0xFEFFFFFF
; CHECK-NEXT: kmovq
    %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $25, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $39, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-33554433, %rax # imm = 0xFDFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $24, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $38, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-67108865, %rax # imm = 0xFBFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $27, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $37, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-134217729, %rax # imm = 0xF7FFFFFF
; CHECK-NEXT: kmovq %rax, %k2

```

```

; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $26, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq
$36, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-268435457, %rax # imm = 0xFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $29, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $35, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-536870913, %rax # imm = 0xDFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $28, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $34, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movq $-1073741825, %rax # imm = 0xBFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k1
; CHECK-NEXT: kshiftrd $31, %k0, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $33, %k2, %k2
; CHECK-NEXT: korq %k2, %k1, %k1
; CHECK-NEXT: movabsq $-2147483649, %rax # imm = 0xFFFFFFFF7FFFFFFF
; CHECK-NEXT:
kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k1, %k2
; CHECK-NEXT: vptestnmb %ymm0, %ymm0, %k1
; CHECK-NEXT: kshiftrd $30, %k0, %k0
; CHECK-NEXT: kshiftrlq $63, %k0, %k0
; CHECK-NEXT: kshiftrq $32, %k0, %k0
; CHECK-NEXT: korq %k0, %k2, %k0
; CHECK-NEXT: movabsq $-4294967297, %rax # imm = 0xFFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $1, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $31, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-8589934593, %rax # imm = 0xFFFFFFFFDFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrlq $63, %k1, %k2
; CHECK-NEXT: kshiftrq $30, %k2, %k2

```

```

; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-17179869185, %rax # imm = 0xFFFFFFFFBFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $3, %k1, %k2
; CHECK-NEXT:
    kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $29, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-34359738369, %rax # imm = 0xFFFFFFFF7FFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $2, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $28, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-68719476737, %rax # imm = 0xFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $5, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $27, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-137438953473, %rax # imm = 0xFFFFFFFFDFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $4, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $26, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
;
CHECK-NEXT: movabsq $-274877906945, %rax # imm = 0xFFFFFFFFBFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $7, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $25, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-549755813889, %rax # imm = 0xFFFFFFFF7FFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $6, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $24, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-1099511627777, %rax # imm = 0xFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0

```

```

; CHECK-NEXT: kshiftrd $9, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $23, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-2199023255553, %rax # imm = 0xFFFFFDFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
;
CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $8, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $22, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-4398046511105, %rax # imm = 0xFFFFFBFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $11, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $21, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-8796093022209, %rax # imm = 0xFFFFF7FFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $10, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $20, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-17592186044417, %rax # imm = 0xFFFFFEFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $13, %k1, %k2
; CHECK-NEXT: kshiftrlq
$63, %k2, %k2
; CHECK-NEXT: kshiftrq $19, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-35184372088833, %rax # imm = 0xFFFFDFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $12, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $18, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-70368744177665, %rax # imm = 0xFFFFBFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $15, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $17, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0

```



```

; CHECK-NEXT: movabsq $-140737488355329, %rax # imm = 0xFFFF7FFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $14, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $16, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT:
    movabsq $-281474976710657, %rax # imm = 0xFFFEFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $17, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $15, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-562949953421313, %rax # imm = 0xFFFDFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $16, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $14, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-1125899906842625, %rax # imm = 0xFFFBFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $19, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $13, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-2251799813685249, %rax # imm = 0xFFF7FFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq
    %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $18, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $12, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-4503599627370497, %rax # imm = 0xFFEFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $21, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $11, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-9007199254740993, %rax # imm = 0xFFDFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $20, %k1, %k2

```

```

; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $10, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-18014398509481985, %rax # imm = 0xFFBFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $23, %k1, %k2
;
CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $9, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-36028797018963969, %rax # imm = 0xFF7FFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $22, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $8, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-72057594037927937, %rax # imm = 0xFEFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $25, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $7, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-144115188075855873, %rax # imm = 0xFDFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $24, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $6, %k2, %k2
; CHECK-NEXT:
korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-288230376151711745, %rax # imm = 0xFBFFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $27, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $5, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-576460752303423489, %rax # imm = 0xF7FFFFFFFFFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $26, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $4, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-1152921504606846977, %rax # imm = 0xEEFFFFFFFFFFFFFFF

```

```

; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $29, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $3, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-2305843009213693953, %rax # imm =
0xDFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $28, %k1, %k2
; CHECK-NEXT: kshiftrlq $63, %k2, %k2
; CHECK-NEXT: kshiftrq $2, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: movabsq $-4611686018427387905, %rax # imm = 0xBFFFFFFF
; CHECK-NEXT: kmovq %rax, %k2
; CHECK-NEXT: kandq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $31, %k1, %k2
; CHECK-NEXT: kshiftrlq $62, %k2, %k2
; CHECK-NEXT: korq %k2, %k0, %k0
; CHECK-NEXT: kshiftrd $30, %k1, %k1
; CHECK-NEXT: kshiftrlq $1, %k0, %k0
; CHECK-NEXT: kshiftrq $1, %k0, %k0
; CHECK-NEXT: kshiftrlq $63, %k1, %k1
; CHECK-NEXT: korq %k1, %k0, %k1
; CHECK-NEXT: vmovdqu8 %ymm1, (%rsi) {%k1}
; CHECK-NEXT: kshiftrq $32, %k1, %k1
; CHECK-NEXT: vmovdqu8 %ymm0, 32(%rsi) {%k1}
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
entry:
%a = load <64 x i8>, <64 x i8>* %x
%b = icmp eq <64 x i8> %a, zeroinitializer
%shuf = shufflevector
<64 x i1> %b, <64 x i1> undef, <64 x i32> <i32 1, i32 0, i32 3, i32 2, i32 5, i32 4, i32 7, i32 6, i32 9, i32 8, i32 11,
i32 10, i32 13, i32 12, i32 15, i32 14, i32 17, i32 16, i32 19, i32 18, i32 21, i32 20, i32 23, i32 22, i32 25, i32 24, i32
27, i32 26, i32 29, i32 28, i32 31, i32 30, i32 33, i32 32, i32 35, i32 34, i32 37, i32 36, i32 39, i32 38, i32 41, i32 40,
i32 43, i32 42, i32 45, i32 44, i32 47, i32 46, i32 49, i32 48, i32 51, i32 50, i32 53, i32 52, i32 55, i32 54, i32 57, i32
56, i32 59, i32 58, i32 61, i32 60, i32 63, i32 62>
call void @llvm.masked.store.v64i8.p0v64i8(<64 x i8> %a, <64 x i8>* %y, i32 1, <64 x i1> %shuf)
ret void
}
declare void @llvm.masked.store.v64i8.p0v64i8(<64 x i8>, <64 x i8>*, i32, <64 x i1>)

@mem64_dst = dso_local global i64 0, align 8
@mem64_src = dso_local global i64 0, align 8
define dso_local i32 @v64i1_inline_asm() "min-legal-vector-width"="256" {
; CHECK-LABEL: v64i1_inline_asm:

```

```

; CHECK:      # %bb.0:
; CHECK-NEXT:  kmovq mem64_src(%rip),
               %k0
; CHECK-NEXT:  #APP
; CHECK-NEXT:  #NO_APP
; CHECK-NEXT:  kmovq %k0, mem64_dst(%rip)
; CHECK-NEXT:  movl -{[0-9]+}}(%rsp), %eax
; CHECK-NEXT:  retq
%1 = alloca i32, align 4
%2 = load i64, i64* @mem64_src, align 8
%3 = call i64 @asm "", "=k,k,~{dirflag},~{fpsr},~{flags}"(i64 %2)
store i64 %3, i64* @mem64_dst, align 8
%4 = load i32, i32* %1, align 4
ret i32 %4
}

```

```

define dso_local void @cmp_v8i64_sext(<8 x i64>* %xptr, <8 x i64>* %yptr, <8 x i64>* %zptr) "min-legal-
vector-width"="256" {

```

```

; CHECK-LABEL: cmp_v8i64_sext:
; CHECK:      # %bb.0:
; CHECK-NEXT:  vmovdqa (%rsi), %ymm0
; CHECK-NEXT:  vmovdqa 32(%rsi), %ymm1
; CHECK-NEXT:  vpcmpgtq 32(%rdi), %ymm1, %ymm1
; CHECK-NEXT:  vpcmpgtq (%rdi), %ymm0, %ymm0
; CHECK-NEXT:  vmovdqa %ymm0, (%rdx)
; CHECK-NEXT:  vmovdqa %ymm1, 32(%rdx)
; CHECK-NEXT:  vzeroupper
; CHECK-NEXT:  retq
%x = load <8 x i64>, <8 x i64>* %xptr
%y = load <8 x i64>, <8 x i64>* %yptr
%cmp = icmp slt <8 x i64> %x, %y
%ext
= sext <8 x i1> %cmp to <8 x i64>
store <8 x i64> %ext, <8 x i64>* %zptr
ret void
}

```

```

define dso_local void @cmp_v8i64_zext(<8 x i64>* %xptr, <8 x i64>* %yptr, <8 x i64>* %zptr) "min-legal-
vector-width"="256" {

```

```

; CHECK-LABEL: cmp_v8i64_zext:
; CHECK:      # %bb.0:
; CHECK-NEXT:  vmovdqa (%rsi), %ymm0
; CHECK-NEXT:  vmovdqa 32(%rsi), %ymm1
; CHECK-NEXT:  vpcmpgtq 32(%rdi), %ymm1, %ymm1
; CHECK-NEXT:  vpcmpgtq (%rdi), %ymm0, %ymm0
; CHECK-NEXT:  vpsrlq $63, %ymm1, %ymm1
; CHECK-NEXT:  vpsrlq $63, %ymm0, %ymm0
; CHECK-NEXT:  vmovdqa %ymm0, (%rdx)

```

```

; CHECK-NEXT: vmovdqa %ymm1, 32(%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%x = load <8 x i64>, <8 x i64>* %xptra
%y = load <8 x i64>, <8 x i64>* %yptra
%cmp = icmp slt <8 x i64> %x, %y
%ext = zext <8 x i1> %cmp to <8 x i64>
store <8 x i64> %ext, <8 x i64>* %zptr
ret void
}

define <16 x i8> @var_rotate_v16i8(<16 x i8> %a, <16 x i8> %b) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: var_rotate_v16i8:
;
CHECK:    # %bb.0:
; CHECK-NEXT: vpandd {{\.?LCPI[0-9]+_[0-9]+}}(%rip){1to4}, %xmm1, %xmm1
; CHECK-NEXT: vpxor %xmm2, %xmm2, %xmm2
; CHECK-NEXT: vpunpckhbw {{\.*#+}} %xmm2 =
xmm1[8],xmm2[8],xmm1[9],xmm2[9],xmm1[10],xmm2[10],xmm1[11],xmm2[11],xmm1[12],xmm2[12],xmm1[13],
xmm2[13],xmm1[14],xmm2[14],xmm1[15],xmm2[15]
; CHECK-NEXT: vpunpckhbw {{\.*#+}} %xmm3 = xmm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15]
; CHECK-NEXT: vpsllvw %xmm2, %xmm3, %xmm2
; CHECK-NEXT: vpsrlw $8, %xmm2, %xmm2
; CHECK-NEXT: vpunpcklbw {{\.*#+}} %xmm0 = xmm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7]
; CHECK-NEXT: vpmovzxbw {{\.*#+}} %xmm1 =
xmm1[0],zero,xmm1[1],zero,xmm1[2],zero,xmm1[3],zero,xmm1[4],zero,xmm1[5],zero,xmm1[6],zero,xmm1[7],zero
; CHECK-NEXT: vpsllvw %xmm1, %xmm0, %xmm0
; CHECK-NEXT: vpsrlw $8, %xmm0, %xmm0
; CHECK-NEXT: vpackuswb %xmm2, %xmm0, %xmm0
; CHECK-NEXT: retq
%b8 = sub <16 x i8> <i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8>, %b
%shl = shl <16 x i8> %a, %b8
%lshr = lshr <16 x i8> %a, %b8
%or = or <16 x i8> %shl, %lshr
ret <16 x i8> %or
}

define <32 x i8> @var_rotate_v32i8(<32 x i8> %a, <32 x i8> %b) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: var_rotate_v32i8:
; CHECK:    # %bb.0:
; CHECK-NEXT: vpandd {{\.?LCPI[0-9]+_[0-9]+}}(%rip){1to8}, %ymm1, %ymm1
; CHECK-NEXT: vpxor %ymm2, %ymm2, %ymm2
; CHECK-NEXT: vpunpckhbw {{\.*#+}} %ymm3 =
ymm1[8],ymm2[8],ymm1[9],ymm2[9],ymm1[10],ymm2[10],ymm1[11],ymm2[11],ymm1[12],ymm2[12],ymm1[13],ymm2[13],ymm1[14],ymm2[14],ymm1[15],ymm2[15],ymm1[16],ymm2[16],ymm1[17],ymm2[17],ymm1[18],ymm2[18],ymm1[19],ymm2[19],ymm1[20],ymm2[20],ymm1[21],ymm2[21],ymm1[22],ymm2[22],ymm1[23],ymm2[23],ymm1[24],ymm2[24],ymm1[25],ymm2[25],ymm1[26],ymm2[26],ymm1[27],ymm2[27],ymm1[28],ymm2[28],ymm1[29],ymm2[29],ymm1[30],ymm2[30],ymm1[31],ymm2[31],ymm1[32],ymm2[32]

```

[illegible]

```

define <32 x i8> @constant_rotate_v32i8(<32 x i8> %a) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: constant_rotate_v32i8:
; CHECK:      # %bb.0:
; CHECK-NEXT:  vpunpckhbw {{.*#+}} ymm1 =
ymm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-NEXT:  vpsllvw {{\.?LCPI[0-9]+\_[0-9]+\}}(%rip), %ymm1, %ymm1
; CHECK-NEXT:  vpsrlw $8, %ymm1, %ymm1
; CHECK-NEXT:  vpunpcklbw {{.*#+}} ymm0 =
ymm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-NEXT:  vpsllvw {{\.?LCPI[0-9]+\_[0-9]+\}}(%rip), %ymm0, %ymm0
; CHECK-NEXT:  vpsrlw $8, %ymm0, %ymm0
; CHECK-NEXT:  vpackuswb %ymm1, %ymm0, %ymm0
; CHECK-NEXT:  retq
%shl = shl <32 x i8> %a, <i8 0, i8 1, i8 2, i8 3, i8 4, i8 5, i8 6, i8 7, i8 8, i8 7, i8 6, i8 5, i8 4, i8 3, i8 2, i8 1, i8 0, i8
1, i8 2, i8 3, i8 4, i8 5, i8 6, i8 7, i8
8, i8 7, i8 6, i8 5, i8 4, i8 3, i8 2, i8 1>
%lshr = lshr <32 x i8> %a, <i8 8, i8 7, i8 6, i8 5, i8 4, i8 3, i8 2, i8 1, i8 0, i8 1, i8 2, i8 3, i8 4, i8 5, i8 6, i8 7, i8 8,
i8 7, i8 6, i8 5, i8 4, i8 3, i8 2, i8 1, i8 0, i8 1, i8 2, i8 3, i8 4, i8 5, i8 6, i8 7>
%or = or <32 x i8> %shl, %lshr
ret <32 x i8> %or
}

```

```

define <32 x i8> @splatconstant_rotate_v32i8(<32 x i8> %a) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: splatconstant_rotate_v32i8:
; CHECK:      # %bb.0:
; CHECK-NEXT:  vpsllw $4, %ymm0, %ymm1
; CHECK-NEXT:  vpsrlw $4, %ymm0, %ymm0
; CHECK-NEXT:  vpternlogd $216, {{\.?LCPI[0-9]+\_[0-9]+\}}(%rip){1to8}, %ymm1, %ymm0
; CHECK-NEXT:  retq
%shl = shl <32 x i8> %a, <i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8
4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4>
%lshr = lshr <32 x i8> %a, <i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8
4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4>
%or = or <32 x i8> %shl, %lshr
ret <32 x i8> %or
}

```

```

define <32 x i8> @splatconstant_rotate_mask_v32i8(<32 x i8> %a) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: splatconstant_rotate_mask_v32i8:
; CHECK:      # %bb.0:
; CHECK-NEXT:  vpsllw $4, %ymm0, %ymm1
; CHECK-NEXT:  vpsrlw $4, %ymm0, %ymm0
; CHECK-NEXT:  vpternlogd $216, {{\.?LCPI[0-9]+\_[0-9]+\}}(%rip){1to8}, %ymm1, %ymm0
; CHECK-NEXT:  vpandd {{\.?LCPI[0-9]+\_[0-9]+\}}(%rip){1to8}, %ymm0, %ymm0
; CHECK-NEXT:  retq
%shl = shl <32 x i8> %a, <i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8
4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4>

```

Copyright 2008, Google Inc.  
All rights reserved.

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

```
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefixes=CHECK,RV64I
; RUN: llc -mtriple=riscv64 -mattr=+zbs -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefixes=CHECK,RV64ZBS
```

## Open Source Used In Debug-container elpaso 3048



```

; RV64I-LABEL: bclr_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sllw a1, a2, a1
; RV64I-NEXT:  not a1, a1
; RV64I-NEXT:  and a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bclr_i32:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  andi a1, a1, 31
; RV64ZBS-NEXT:  bclr a0, a0, a1
; RV64ZBS-NEXT:  sext.w a0, a0
; RV64ZBS-NEXT:  ret
%and = and i32 %b, 31
%shl = shl nuw i32 1, %and
%neg = xor i32 %shl, -1
%and1 = and i32 %neg, %a
ret i32 %and1
}

```

```

define signext i32 @bclr_i32_no_mask(i32 signext %a,
i32 signext %b) nounwind {

```

```

; RV64I-LABEL: bclr_i32_no_mask:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sllw a1, a2, a1
; RV64I-NEXT:  not a1, a1
; RV64I-NEXT:  and a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bclr_i32_no_mask:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bclr a0, a0, a1
; RV64ZBS-NEXT:  sext.w a0, a0
; RV64ZBS-NEXT:  ret
%shl = shl i32 1, %b
%neg = xor i32 %shl, -1
%and1 = and i32 %neg, %a
ret i32 %and1
}

```

```

define signext i32 @bclr_i32_load(ptr %p, i32 signext %b) nounwind {

```

```

; RV64I-LABEL: bclr_i32_load:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lw a0, 0(a0)
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sllw a1, a2, a1

```

```

; RV64I-NEXT:  not a1, a1
; RV64I-NEXT:  and a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bclr_i32_load:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  lw a0, 0(a0)
; RV64ZBS-NEXT:  bclr a0, a0, a1
; RV64ZBS-NEXT:  sext.w a0, a0
; RV64ZBS-NEXT:  ret
%a = load i32, ptr %p
%shl = shl i32 1, %b
%neg
= xor i32 %shl, -1
%and1 = and i32 %neg, %a
ret i32 %and1
}

```

```

define i64 @bclr_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: bclr_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sll a1, a2, a1
; RV64I-NEXT:  not a1, a1
; RV64I-NEXT:  and a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bclr_i64:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  bclr a0, a0, a1
; RV64ZBS-NEXT:  ret
%and = and i64 %b, 63
%shl = shl nuw i64 1, %and
%neg = xor i64 %shl, -1
%and1 = and i64 %neg, %a
ret i64 %and1
}

```

```

define i64 @bclr_i64_no_mask(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: bclr_i64_no_mask:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sll a1, a2, a1
; RV64I-NEXT:  not a1, a1
; RV64I-NEXT:  and a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bclr_i64_no_mask:

```

```

; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bclr a0, a0, a1
; RV64ZBS-NEXT:  ret
%shl = shl i64 1, %b
%neg = xor i64 %shl, -1
%and1 = and i64 %neg, %a
ret i64
%and1
}

```

```

define signext i32 @bset_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: bset_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sllw a1, a2, a1
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bset_i32:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  andi a1, a1, 31
; RV64ZBS-NEXT:  bset a0, a0, a1
; RV64ZBS-NEXT:  sext.w a0, a0
; RV64ZBS-NEXT:  ret
%and = and i32 %b, 31
%shl = shl nuw i32 1, %and
%or = or i32 %shl, %a
ret i32 %or
}

```

```

define signext i32 @bset_i32_no_mask(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: bset_i32_no_mask:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sllw a1, a2, a1
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bset_i32_no_mask:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bset a0, a0, a1
; RV64ZBS-NEXT:  sext.w a0, a0
; RV64ZBS-NEXT:  ret
%shl = shl i32 1, %b
%or = or i32 %shl, %a
ret i32 %or
}

```

```

define signext i32
@bset_i32_load(ptr %p, i32 signext %b) nounwind {
; RV64I-LABEL: bset_i32_load:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lw a0, 0(a0)
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sllw a1, a2, a1
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bset_i32_load:
; RV64ZBS:     # %bb.0:
; RV64ZBS-NEXT: lw a0, 0(a0)
; RV64ZBS-NEXT: bset a0, a0, a1
; RV64ZBS-NEXT: sext.w a0, a0
; RV64ZBS-NEXT: ret
%a = load i32, ptr %p
%shl = shl i32 1, %b
%or = or i32 %shl, %a
ret i32 %or
}

```

; We can use bsetw for  $1 \ll x$  by setting the first source to zero.

```

define signext i32 @bset_i32_zero(i32 signext %a) nounwind {
; RV64I-LABEL: bset_i32_zero:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 1
; RV64I-NEXT:  sllw a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bset_i32_zero:
; RV64ZBS:     # %bb.0:
; RV64ZBS-NEXT: bset a0, zero, a0
; RV64ZBS-NEXT: sext.w a0, a0
; RV64ZBS-NEXT: ret
%shl = shl i32 1, %a
ret i32 %shl
}

```

```

define i64 @bset_i64(i64 %a,
i64 %b) nounwind {
; RV64I-LABEL: bset_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sll a1, a2, a1
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;

```

```

; RV64ZBS-LABEL: bset_i64:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bset a0, a0, a1
; RV64ZBS-NEXT:  ret
%conv = and i64 %b, 63
%shl = shl nuw i64 1, %conv
%or = or i64 %shl, %a
ret i64 %or
}

```

```

define i64 @bset_i64_no_mask(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: bset_i64_no_mask:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sll a1, a2, a1
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bset_i64_no_mask:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bset a0, a0, a1
; RV64ZBS-NEXT:  ret
%shl = shl i64 1, %b
%or = or i64 %shl, %a
ret i64 %or
}

```

; We can use bsetw for  $1 \ll x$  by setting the first source to zero.

```

define signext i64 @bset_i64_zero(i64 signext %a) nounwind {
; RV64I-LABEL: bset_i64_zero:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li
a1, 1
; RV64I-NEXT:  sll a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bset_i64_zero:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bset a0, zero, a0
; RV64ZBS-NEXT:  ret
%shl = shl i64 1, %a
ret i64 %shl
}

```

```

define signext i32 @binv_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: binv_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a2, 1

```

```

; RV64I-NEXT:  sllw a1, a2, a1
; RV64I-NEXT:  xor a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: binv_i32:
; RV64ZBS:     # %bb.0:
; RV64ZBS-NEXT:  andi a1, a1, 31
; RV64ZBS-NEXT:  binv a0, a0, a1
; RV64ZBS-NEXT:  sext.w a0, a0
; RV64ZBS-NEXT:  ret
%and = and i32 %b, 31
%shl = shl nuw i32 1, %and
%xor = xor i32 %shl, %a
ret i32 %xor
}

define signext i32 @binv_i32_no_mask(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: binv_i32_no_mask:
; RV64I:     # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sllw a1, a2, a1
; RV64I-NEXT:  xor a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: binv_i32_no_mask:
;
; RV64ZBS:     # %bb.0:
; RV64ZBS-NEXT:  binv a0, a0, a1
; RV64ZBS-NEXT:  sext.w a0, a0
; RV64ZBS-NEXT:  ret
%shl = shl i32 1, %b
%xor = xor i32 %shl, %a
ret i32 %xor
}

define signext i32 @binv_i32_load(ptr %p, i32 signext %b) nounwind {
; RV64I-LABEL: binv_i32_load:
; RV64I:     # %bb.0:
; RV64I-NEXT:  lw a0, 0(a0)
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sllw a1, a2, a1
; RV64I-NEXT:  xor a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: binv_i32_load:
; RV64ZBS:     # %bb.0:
; RV64ZBS-NEXT:  lw a0, 0(a0)
; RV64ZBS-NEXT:  binv a0, a0, a1

```

```

; RV64ZBS-NEXT:  sext.w a0, a0
; RV64ZBS-NEXT:  ret
%a = load i32, ptr %p
%shl = shl i32 1, %b
%xor = xor i32 %shl, %a
ret i32 %xor
}

define i64 @binv_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: binv_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sll a1, a2, a1
; RV64I-NEXT:  xor a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: binv_i64:
; RV64ZBS:    # %bb.0:
;
RV64ZBS-NEXT:  binv a0, a0, a1
; RV64ZBS-NEXT:  ret
%conv = and i64 %b, 63
%shl = shl nuw i64 1, %conv
%xor = xor i64 %shl, %a
ret i64 %xor
}

define i64 @binv_i64_no_mask(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: binv_i64_no_mask:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  sll a1, a2, a1
; RV64I-NEXT:  xor a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: binv_i64_no_mask:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  binv a0, a0, a1
; RV64ZBS-NEXT:  ret
%shl = shl nuw i64 1, %b
%xor = xor i64 %shl, %a
ret i64 %xor
}

define signext i32 @bext_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: bext_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srlw a0, a0, a1

```

```

; RV64I-NEXT:  andi a0, a0, 1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bext_i32:
; RV64ZBS:     # %bb.0:
; RV64ZBS-NEXT:  andi a1, a1, 31
; RV64ZBS-NEXT:  bext a0, a0, a1
; RV64ZBS-NEXT:  ret
%and = and i32 %b, 31
%shr = lshr i32 %a, %and
%and1 = and i32
%shr, 1
ret i32 %and1
}

define signext i32 @bext_i32_no_mask(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: bext_i32_no_mask:
; RV64I:       # %bb.0:
; RV64I-NEXT:  srlw a0, a0, a1
; RV64I-NEXT:  andi a0, a0, 1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bext_i32_no_mask:
; RV64ZBS:     # %bb.0:
; RV64ZBS-NEXT:  bext a0, a0, a1
; RV64ZBS-NEXT:  ret
%shr = lshr i32 %a, %b
%and1 = and i32 %shr, 1
ret i32 %and1
}

; This gets previous converted to (i1 (truncate (srl X, Y))). Make sure we are
; able to use bext.
define void @bext_i32_trunc(i32 signext %0, i32 signext %1) {
; RV64I-LABEL: bext_i32_trunc:
; RV64I:       # %bb.0:
; RV64I-NEXT:  srlw a0, a0, a1
; RV64I-NEXT:  andi a0, a0, 1
; RV64I-NEXT:  beqz a0, .LBB19_2
; RV64I-NEXT: # %bb.1:
; RV64I-NEXT:  ret
; RV64I-NEXT: .LBB19_2:
; RV64I-NEXT:  tail bar
;
; RV64ZBS-LABEL: bext_i32_trunc:
; RV64ZBS:     # %bb.0:
; RV64ZBS-NEXT:  bext a0, a0, a1
; RV64ZBS-NEXT:  beqz a0, .LBB19_2

```



```

; RV64ZBS-NEXT:
    # %bb.1:
; RV64ZBS-NEXT:  ret
; RV64ZBS-NEXT: .LBB19_2:
; RV64ZBS-NEXT:  tail bar
    %3 = shl i32 1, %1
    %4 = and i32 %3, %0
    %5 = icmp eq i32 %4, 0
    br i1 %5, label %6, label %7

6:                                     ; preds = %2
    tail call void @bar()
    br label %7

7:                                     ; preds = %6, %2
    ret void
}

declare void @bar()

define i64 @bext_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: bext_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srl a0, a0, a1
; RV64I-NEXT:  andi a0, a0, 1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bext_i64:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bext a0, a0, a1
; RV64ZBS-NEXT:  ret
    %conv = and i64 %b, 63
    %shr = lshr i64 %a, %conv
    %and1 = and i64 %shr, 1
    ret i64 %and1
}

define i64 @bext_i64_no_mask(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: bext_i64_no_mask:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srl a0, a0, a1
; RV64I-NEXT:  andi a0, a0, 1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL:
    bext_i64_no_mask:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bext a0, a0, a1

```

```

; RV64ZBS-NEXT:  ret
%shr = lshr i64 %a, %b
%and1 = and i64 %shr, 1
ret i64 %and1
}

define signext i32 @bexti_i32(i32 signext %a) nounwind {
; RV64I-LABEL: bexti_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 58
; RV64I-NEXT:  srli a0, a0, 63
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bexti_i32:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bexti a0, a0, 5
; RV64ZBS-NEXT:  ret
%shr = lshr i32 %a, 5
%and = and i32 %shr, 1
ret i32 %and
}

define i64 @bexti_i64(i64 %a) nounwind {
; RV64I-LABEL: bexti_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 58
; RV64I-NEXT:  srli a0, a0, 63
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bexti_i64:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bexti a0, a0, 5
; RV64ZBS-NEXT:  ret
%shr = lshr i64 %a, 5
%and = and i64 %shr, 1
ret i64 %and
}

define signext i32 @bexti_i32_cmp(i32 signext %a) nounwind {
; CHECK-LABEL: bexti_i32_cmp:
;
CHECK:      # %bb.0:
; CHECK-NEXT:  andi a0, a0, 32
; CHECK-NEXT:  snez a0, a0
; CHECK-NEXT:  ret
%and = and i32 %a, 32
%cmp = icmp ne i32 %and, 0
%zext = zext i1 %cmp to i32

```

```

ret i32 %zext
}

define i64 @bexti_i64_cmp(i64 %a) nounwind {
; RV64I-LABEL: bexti_i64_cmp:
; RV64I:      # %bb.0:
; RV64I-NEXT: slli a0, a0, 58
; RV64I-NEXT: srli a0, a0, 63
; RV64I-NEXT: ret
;
; RV64ZBS-LABEL: bexti_i64_cmp:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT: bexti a0, a0, 5
; RV64ZBS-NEXT: ret
%and = and i64 %a, 32
%cmp = icmp ne i64 %and, 0
%zext = zext i1 %cmp to i64
ret i64 %zext
}

define signext i32 @bclri_i32_10(i32 signext %a) nounwind {
; CHECK-LABEL: bclri_i32_10:
; CHECK:      # %bb.0:
; CHECK-NEXT: andi a0, a0, -1025
; CHECK-NEXT: ret
%and = and i32 %a, -1025
ret i32 %and
}

define signext i32 @bclri_i32_11(i32 signext %a) nounwind {
; RV64I-LABEL: bclri_i32_11:
; RV64I:      # %bb.0:
; RV64I-NEXT: lui a1, 1048575
; RV64I-NEXT: addiw a1,
a1, 2047
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: ret
;
; RV64ZBS-LABEL: bclri_i32_11:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT: bclri a0, a0, 11
; RV64ZBS-NEXT: ret
%and = and i32 %a, -2049
ret i32 %and
}

define signext i32 @bclri_i32_30(i32 signext %a) nounwind {
; RV64I-LABEL: bclri_i32_30:

```

```

; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a1, 786432
; RV64I-NEXT:  addiw a1, a1, -1
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bclri_i32_30:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bclri a0, a0, 30
; RV64ZBS-NEXT:  ret
%and = and i32 %a, -1073741825
ret i32 %and
}

```

```

define signext i32 @bclri_i32_31(i32 signext %a) nounwind {
; RV64I-LABEL: bclri_i32_31:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 33
; RV64I-NEXT:  srli a0, a0, 33
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bclri_i32_31:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bclri a0, a0, 31
; RV64ZBS-NEXT:  sext.w a0, a0
; RV64ZBS-NEXT:  ret
%and = and i32 %a,
-2147483649
ret i32 %and
}

```

```

define i64 @bclri_i64_10(i64 %a) nounwind {
; CHECK-LABEL: bclri_i64_10:
; CHECK:    # %bb.0:
; CHECK-NEXT:  andi a0, a0, -1025
; CHECK-NEXT:  ret
%and = and i64 %a, -1025
ret i64 %and
}

```

```

define i64 @bclri_i64_11(i64 %a) nounwind {
; RV64I-LABEL: bclri_i64_11:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a1, 1048575
; RV64I-NEXT:  addiw a1, a1, 2047
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  ret
;

```

```

; RV64ZBS-LABEL: bclri_i64_11:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bclri a0, a0, 11
; RV64ZBS-NEXT:  ret
%and = and i64 %a, -2049
ret i64 %and
}

define i64 @bclri_i64_30(i64 %a) nounwind {
; RV64I-LABEL: bclri_i64_30:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a1, 786432
; RV64I-NEXT:  addiw a1, a1, -1
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bclri_i64_30:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bclri a0, a0, 30
; RV64ZBS-NEXT:  ret
%and = and i64 %a, -1073741825
ret i64 %and
}

define
i64 @bclri_i64_31(i64 %a) nounwind {
; RV64I-LABEL: bclri_i64_31:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a1, 524288
; RV64I-NEXT:  addi a1, a1, -1
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bclri_i64_31:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bclri a0, a0, 31
; RV64ZBS-NEXT:  ret
%and = and i64 %a, -2147483649
ret i64 %and
}

define i64 @bclri_i64_62(i64 %a) nounwind {
; RV64I-LABEL: bclri_i64_62:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a1, -1
; RV64I-NEXT:  slli a1, a1, 62
; RV64I-NEXT:  addi a1, a1, -1
; RV64I-NEXT:  and a0, a0, a1

```

```

; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bclri_i64_62:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  bclri a0, a0, 62
; RV64ZBS-NEXT:  ret
%and = and i64 %a, -4611686018427387905
ret i64 %and
}

```

```

define i64 @bclri_i64_63(i64 %a) nounwind {
; RV64I-LABEL: bclri_i64_63:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 1
; RV64I-NEXT:  srli a0, a0, 1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL:
bclri_i64_63:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  bclri a0, a0, 63
; RV64ZBS-NEXT:  ret
%and = and i64 %a, -9223372036854775809
ret i64 %and
}

```

```

define i64 @bclri_i64_large0(i64 %a) nounwind {
; RV64I-LABEL: bclri_i64_large0:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a1, 1044480
; RV64I-NEXT:  addiw a1, a1, -256
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bclri_i64_large0:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  andi a0, a0, -256
; RV64ZBS-NEXT:  bclri a0, a0, 24
; RV64ZBS-NEXT:  ret
%and = and i64 %a, -16777472
ret i64 %and
}

```

```

define i64 @bclri_i64_large1(i64 %a) nounwind {
; RV64I-LABEL: bclri_i64_large1:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a1, 1044464
; RV64I-NEXT:  addiw a1, a1, -1

```

```

; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bclri_i64_large1:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  bclri a0, a0, 16
; RV64ZBS-NEXT:  bclri a0, a0, 24
; RV64ZBS-NEXT:  ret
%and = and i64 %a,
-16842753
ret i64 %and
}

```

```

define signext i32 @bseti_i32_10(i32 signext %a) nounwind {
; CHECK-LABEL: bseti_i32_10:
; CHECK:      # %bb.0:
; CHECK-NEXT:  ori a0, a0, 1024
; CHECK-NEXT:  ret
%or = or i32 %a, 1024
ret i32 %or
}

```

```

define signext i32 @bseti_i32_11(i32 signext %a) nounwind {
; RV64I-LABEL: bseti_i32_11:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 1
; RV64I-NEXT:  slliw a1, a1, 11
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bseti_i32_11:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  bseti a0, a0, 11
; RV64ZBS-NEXT:  ret
%or = or i32 %a, 2048
ret i32 %or
}

```

```

define signext i32 @bseti_i32_30(i32 signext %a) nounwind {
; RV64I-LABEL: bseti_i32_30:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a1, 262144
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bseti_i32_30:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  bseti a0, a0, 30

```

```

; RV64ZBS-NEXT:  ret
%or = or i32 %a, 1073741824
ret i32 %or
}

define signext i32
@bseti_i32_31(i32 signext %a) nounwind {
; RV64I-LABEL: bseti_i32_31:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a1, 524288
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bseti_i32_31:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bseti a0, a0, 31
; RV64ZBS-NEXT:  sext.w a0, a0
; RV64ZBS-NEXT:  ret
%or = or i32 %a, 2147483648
ret i32 %or
}

define i64 @bseti_i64_10(i64 %a) nounwind {
; CHECK-LABEL: bseti_i64_10:
; CHECK:      # %bb.0:
; CHECK-NEXT:  ori a0, a0, 1024
; CHECK-NEXT:  ret
%or = or i64 %a, 1024
ret i64 %or
}

define i64 @bseti_i64_11(i64 %a) nounwind {
; RV64I-LABEL: bseti_i64_11:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 1
; RV64I-NEXT:  slli a1, a1, 11
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bseti_i64_11:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bseti a0, a0, 11
; RV64ZBS-NEXT:  ret
%or = or i64 %a, 2048
ret i64 %or
}

define i64 @bseti_i64_30(i64 %a) nounwind {

```



```

; RV64I-LABEL: bseti_i64_30:
;
RV64I:    # %bb.0:
; RV64I-NEXT:  lui a1, 262144
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bseti_i64_30:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bseti a0, a0, 30
; RV64ZBS-NEXT:  ret
%or = or i64 %a, 1073741824
ret i64 %or
}

define i64 @bseti_i64_31(i64 %a) nounwind {
; RV64I-LABEL: bseti_i64_31:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a1, 1
; RV64I-NEXT:  slli a1, a1, 31
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bseti_i64_31:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bseti a0, a0, 31
; RV64ZBS-NEXT:  ret
%or = or i64 %a, 2147483648
ret i64 %or
}

define i64 @bseti_i64_62(i64 %a) nounwind {
; RV64I-LABEL: bseti_i64_62:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a1, 1
; RV64I-NEXT:  slli a1, a1, 62
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bseti_i64_62:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bseti a0, a0, 62
; RV64ZBS-NEXT:  ret
%or = or i64 %a,
4611686018427387904
ret i64 %or
}

```

```

define i64 @bseti_i64_63(i64 %a) nounwind {
; RV64I-LABEL: bseti_i64_63:
; RV64I:      # %bb.0:
; RV64I-NEXT: li a1, -1
; RV64I-NEXT: slli a1, a1, 63
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: ret
;
; RV64ZBS-LABEL: bseti_i64_63:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT: bseti a0, a0, 63
; RV64ZBS-NEXT: ret
%or = or i64 %a, 9223372036854775808
ret i64 %or
}

```

```

define signext i32 @binvi_i32_10(i32 signext %a) nounwind {
; CHECK-LABEL: binvi_i32_10:
; CHECK:      # %bb.0:
; CHECK-NEXT: xori a0, a0, 1024
; CHECK-NEXT: ret
%xor = xor i32 %a, 1024
ret i32 %xor
}

```

```

define signext i32 @binvi_i32_11(i32 signext %a) nounwind {
; RV64I-LABEL: binvi_i32_11:
; RV64I:      # %bb.0:
; RV64I-NEXT: li a1, 1
; RV64I-NEXT: slliw a1, a1, 11
; RV64I-NEXT: xor a0, a0, a1
; RV64I-NEXT: ret
;
; RV64ZBS-LABEL: binvi_i32_11:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT: binvi a0, a0, 11
; RV64ZBS-NEXT: ret
%xor = xor i32 %a,
2048
ret i32 %xor
}

```

```

define signext i32 @binvi_i32_30(i32 signext %a) nounwind {
; RV64I-LABEL: binvi_i32_30:
; RV64I:      # %bb.0:
; RV64I-NEXT: lui a1, 262144
; RV64I-NEXT: xor a0, a0, a1
; RV64I-NEXT: ret

```

```

;
; RV64ZBS-LABEL: binvi_i32_30:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  binvi a0, a0, 30
; RV64ZBS-NEXT:  ret
%xor = xor i32 %a, 1073741824
ret i32 %xor
}

```

```

define signext i32 @binvi_i32_31(i32 signext %a) nounwind {
; RV64I-LABEL: binvi_i32_31:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a1, 524288
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: binvi_i32_31:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  binvi a0, a0, 31
; RV64ZBS-NEXT:  sext.w a0, a0
; RV64ZBS-NEXT:  ret
%xor = xor i32 %a, 2147483648
ret i32 %xor
}

```

```

define i64 @binvi_i64_10(i64 %a) nounwind {
; CHECK-LABEL: binvi_i64_10:
; CHECK:    # %bb.0:
; CHECK-NEXT:  xori a0, a0, 1024
; CHECK-NEXT:  ret
%xor = xor i64 %a, 1024
ret i64 %xor
}

```

```

define i64 @binvi_i64_11(i64
%a) nounwind {
; RV64I-LABEL: binvi_i64_11:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a1, 1
; RV64I-NEXT:  slli a1, a1, 11
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: binvi_i64_11:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  binvi a0, a0, 11
; RV64ZBS-NEXT:  ret
%xor = xor i64 %a, 2048

```

```

ret i64 %xor
}

define i64 @binvi_i64_30(i64 %a) nounwind {
; RV64I-LABEL: binvi_i64_30:
; RV64I:      # %bb.0:
; RV64I-NEXT: lui a1, 262144
; RV64I-NEXT: xor a0, a0, a1
; RV64I-NEXT: ret
;
; RV64ZBS-LABEL: binvi_i64_30:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT: binvi a0, a0, 30
; RV64ZBS-NEXT: ret
%xor = xor i64 %a, 1073741824
ret i64 %xor
}

```

```

define i64 @binvi_i64_31(i64 %a) nounwind {
; RV64I-LABEL: binvi_i64_31:
; RV64I:      # %bb.0:
; RV64I-NEXT: li a1, 1
; RV64I-NEXT: slli a1, a1, 31
; RV64I-NEXT: xor a0, a0, a1
; RV64I-NEXT: ret
;
; RV64ZBS-LABEL: binvi_i64_31:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT: binvi
a0, a0, 31
; RV64ZBS-NEXT: ret
%xor = xor i64 %a, 2147483648
ret i64 %xor
}

```

```

define i64 @binvi_i64_62(i64 %a) nounwind {
; RV64I-LABEL: binvi_i64_62:
; RV64I:      # %bb.0:
; RV64I-NEXT: li a1, 1
; RV64I-NEXT: slli a1, a1, 62
; RV64I-NEXT: xor a0, a0, a1
; RV64I-NEXT: ret
;
; RV64ZBS-LABEL: binvi_i64_62:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT: binvi a0, a0, 62
; RV64ZBS-NEXT: ret
%xor = xor i64 %a, 4611686018427387904

```

```

ret i64 %xor
}

define i64 @binvi_i64_63(i64 %a) nounwind {
; RV64I-LABEL: binvi_i64_63:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a1, -1
; RV64I-NEXT:  slli a1, a1, 63
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: binvi_i64_63:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  binvi a0, a0, 63
; RV64ZBS-NEXT:  ret
%xor = xor i64 %a, 9223372036854775808
ret i64 %xor
}

```

```

define i64 @xor_i64_large(i64 %a) nounwind {
; RV64I-LABEL: xor_i64_large:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a1, 1
; RV64I-NEXT:  slli
a1, a1, 32
; RV64I-NEXT:  addi a1, a1, 1
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: xor_i64_large:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  binvi a0, a0, 0
; RV64ZBS-NEXT:  binvi a0, a0, 32
; RV64ZBS-NEXT:  ret
%xor = xor i64 %a, 4294967297
ret i64 %xor
}

```

```

define i64 @xor_i64_4099(i64 %a) nounwind {
; RV64I-LABEL: xor_i64_4099:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a1, 1
; RV64I-NEXT:  addiw a1, a1, 3
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: xor_i64_4099:
; RV64ZBS:    # %bb.0:

```

```

; RV64ZBS-NEXT:  xori a0, a0, 3
; RV64ZBS-NEXT:  binvi a0, a0, 12
; RV64ZBS-NEXT:  ret
%xor = xor i64 %a, 4099
ret i64 %xor
}

```

```

define i64 @xor_i64_96(i64 %a) nounwind {
; CHECK-LABEL: xor_i64_96:
; CHECK:      # %bb.0:
; CHECK-NEXT:  xori a0, a0, 96
; CHECK-NEXT:  ret
%xor = xor i64 %a, 96
ret i64 %xor
}

```

```

define i64 @or_i64_large(i64 %a) nounwind {
; RV64I-LABEL: or_i64_large:
; RV64I:      # %bb.0:
; RV64I-NEXT:
    li a1, 1
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  addi a1, a1, 1
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: or_i64_large:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  bseti a0, a0, 0
; RV64ZBS-NEXT:  bseti a0, a0, 32
; RV64ZBS-NEXT:  ret
%or = or i64 %a, 4294967297
ret i64 %or
}

```

```

define i64 @xor_i64_66901(i64 %a) nounwind {
; RV64I-LABEL: xor_i64_66901:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a1, 16
; RV64I-NEXT:  addiw a1, a1, 1365
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: xor_i64_66901:
; RV64ZBS:    # %bb.0:
; RV64ZBS-NEXT:  xori a0, a0, 1365
; RV64ZBS-NEXT:  binvi a0, a0, 16
; RV64ZBS-NEXT:  ret

```

```

%xor = xor i64 %a, 66901
ret i64 %xor
}

define i64 @or_i64_4099(i64 %a) nounwind {
; RV64I-LABEL: or_i64_4099:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a1, 1
; RV64I-NEXT:  addiw a1, a1, 3
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: or_i64_4099:
; RV64ZBS:
;   # %bb.0:
; RV64ZBS-NEXT:  ori a0, a0, 3
; RV64ZBS-NEXT:  bseti a0, a0, 12
; RV64ZBS-NEXT:  ret
%or = or i64 %a, 4099
ret i64 %or
}

define i64 @or_i64_96(i64 %a) nounwind {
; CHECK-LABEL: or_i64_96:
; CHECK:      # %bb.0:
; CHECK-NEXT:  ori a0, a0, 96
; CHECK-NEXT:  ret
%or = or i64 %a, 96
ret i64 %or
}

define i64 @or_i64_66901(i64 %a) nounwind {
; RV64I-LABEL: or_i64_66901:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a1, 16
; RV64I-NEXT:  addiw a1, a1, 1365
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: or_i64_66901:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  ori a0, a0, 1365
; RV64ZBS-NEXT:  bseti a0, a0, 16
; RV64ZBS-NEXT:  ret
%or = or i64 %a, 66901
ret i64 %or
}

```

```

define signext i32 @bset_trailing_ones_i32_mask(i32 signext %a) nounwind {
; RV64I-LABEL: bset_trailing_ones_i32_mask:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, -1
; RV64I-NEXT:  sllw a0, a1, a0
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  ret
;
;
RV64ZBS-LABEL: bset_trailing_ones_i32_mask:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  andi a0, a0, 31
; RV64ZBS-NEXT:  bset a0, zero, a0
; RV64ZBS-NEXT:  addiw a0, a0, -1
; RV64ZBS-NEXT:  ret
%and = and i32 %a, 31
%shift = shl nsw i32 -1, %and
%not = xor i32 %shift, -1
ret i32 %not
}

define signext i32 @bset_trailing_ones_i32_no_mask(i32 signext %a) nounwind {
; RV64I-LABEL: bset_trailing_ones_i32_no_mask:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, -1
; RV64I-NEXT:  sllw a0, a1, a0
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  ret
;
;
RV64ZBS-LABEL: bset_trailing_ones_i32_no_mask:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  bset a0, zero, a0
; RV64ZBS-NEXT:  addiw a0, a0, -1
; RV64ZBS-NEXT:  ret
%shift = shl nsw i32 -1, %a
%not = xor i32 %shift, -1
ret i32 %not
}

define signext i64 @bset_trailing_ones_i64_mask(i64 signext %a) nounwind {
; RV64I-LABEL: bset_trailing_ones_i64_mask:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, -1
; RV64I-NEXT:  sll
a0, a1, a0
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  ret
;
;

```



```

; RV64ZBS-LABEL: bset_trailing_ones_i64_mask:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  bset a0, zero, a0
; RV64ZBS-NEXT:  addi a0, a0, -1
; RV64ZBS-NEXT:  ret
%and = and i64 %a, 63
%shift = shl nsw i64 -1, %and
%not = xor i64 %shift, -1
ret i64 %not
}

define signext i64 @bset_trailing_ones_i64_no_mask(i64 signext %a) nounwind {
; RV64I-LABEL: bset_trailing_ones_i64_no_mask:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, -1
; RV64I-NEXT:  sll a0, a1, a0
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  ret
;
; RV64ZBS-LABEL: bset_trailing_ones_i64_no_mask:
; RV64ZBS:      # %bb.0:
; RV64ZBS-NEXT:  bset a0, zero, a0
; RV64ZBS-NEXT:  addi a0, a0, -1
; RV64ZBS-NEXT:  ret
%shift = shl nsw i64 -1, %a
%not = xor i64 %shift, -1
ret i64 %not
}

```

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
 Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

## APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

=====  
Software from third parties included in the LLVM Project:  
=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or `LICENSE` file at the top containing the specific license and restrictions which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every file.

=====  
Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):  
=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2003-2019 University of Illinois at Urbana-Champaign.  
All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

<http://llvm.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- \* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but



excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any

risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

=====

Software from third parties included in the LLVM Project:

=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or `LICENSE` file at the top containing the specific license and restrictions which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every file.

=====

Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):

=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2010 Apple Inc.  
All rights reserved.

Developed by:

LLDB Team

<http://lldb.lvm.org/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- \* Neither the names of the LLDB Team, copyright holders, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

-----  
clang-tidy High-Integrity C++ Files  
-----

All clang-tidy files are licensed under the same terms as the rest of the LLVM project with the following additions:

Any file referencing a High-Integrity C++ Coding guideline:

HIC++ Coding Standard as created by PRQA.

Please see

<https://www.perforce.com/resources/qac/high-integrity-cpp-coding-standard/conditions-of-use>  
for more information.

ISC LICENSE

This license is approved by the OSI and FSF as GPL-compatible.

<http://opensource.org/licenses/isc-license.txt>

Copyright (c) 2013-2014, Pexpect development team  
Copyright (c) 2012, Noah Spurrier <noah@noah.org>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

```
; RUN: opt -passes=consthoist -S -o - %s | FileCheck %s
target datalayout = "e-m:e-p:32:32-i64:64-v128:64:128-a:0:32-n32-S64"
target triple = "thumbv6m-none--musleabi"
```

```
; Check that for i8 type, the maximum legal offset is 31.
; Also check that an constant used as value to be stored rather than
; pointer in a store instruction is hoisted.
; CHECK: foo_i8
; CHECK-DAG: %[[C1:const[0-9]?]] = bitcast i32 805874720 to i32
; CHECK-DAG: %[[C2:const[0-9]?]] = bitcast i32 805874688 to i32
; CHECK-DAG: %[[C3:const[0-9]?]] = bitcast i32 805873720 to i32
; CHECK-DAG: %[[C4:const[0-9]?]] = bitcast i32 805873688 to i32
; CHECK: %0 = inttoptr i32 %[[C2]] to ptr
; CHECK-NEXT: %1 = load volatile i8, ptr %0
; CHECK-NEXT: %[[M1:const_mat[0-9]?]] = add i32 %[[C2]], 4
; CHECK-NEXT: %2 = inttoptr i32 %[[M1]] to ptr
; CHECK-NEXT: %3 = load volatile i8, ptr %2
; CHECK-NEXT: %[[M2:const_mat[0-9]?]] = add i32 %[[C2]], 31
; CHECK-NEXT: %4 = inttoptr i32 %[[M2]] to ptr
; CHECK-NEXT: %5 = load volatile
i8, ptr %4
; CHECK-NEXT: %6 = inttoptr i32 %[[C1]] to ptr
; CHECK-NEXT: %7 = load volatile i8, ptr %6
; CHECK-NEXT: %[[M3:const_mat[0-9]?]] = add i32 %[[C1]], 7
; CHECK-NEXT: %8 = inttoptr i32 %[[M3]] to ptr
; CHECK-NEXT: %9 = load volatile i8, ptr %8
; CHECK-NEXT: %10 = inttoptr i32 %[[C4]] to ptr
; CHECK-NEXT: store i8 %9, ptr %10
; CHECK-NEXT: %[[M4:const_mat[0-9]?]] = add i32 %[[C4]], 31
; CHECK-NEXT: %11 = inttoptr i32 %[[M4]] to ptr
; CHECK-NEXT: store i8 %7, ptr %11
; CHECK-NEXT: %12 = inttoptr i32 %[[C3]] to ptr
; CHECK-NEXT: store i8 %5, ptr %12
```

```
; CHECK-NEXT: %[[M5:const_mat[0-9]?]] = add i32 %[[C3]], 7
; CHECK-NEXT: %13 = inttoptr i32 %[[M5]] to ptr
; CHECK-NEXT: store i8 %3, ptr %13
; CHECK-NEXT: %[[M6:const_mat[0-9]?]] = add i32 %[[C1]], 80
; CHECK-NEXT: %14 = inttoptr i32 %[[M6]] to ptr
; CHECK-NEXT: store ptr %14, ptr @goo
```

```
@goo = global ptr undef
```

```
define void @foo_i8() {
entry:
  %0 = load volatile i8, ptr inttoptr (i32 805874688 to ptr)
  %1 = load volatile
i8, ptr inttoptr (i32 805874692 to ptr)
  %2 = load volatile i8, ptr inttoptr (i32 805874719 to ptr)
  %3 = load volatile i8, ptr inttoptr (i32 805874720 to ptr)
  %4 = load volatile i8, ptr inttoptr (i32 805874727 to ptr)
  store i8 %4, ptr inttoptr(i32 805873688 to ptr)
  store i8 %3, ptr inttoptr(i32 805873719 to ptr)
  store i8 %2, ptr inttoptr(i32 805873720 to ptr)
  store i8 %1, ptr inttoptr(i32 805873727 to ptr)
  store ptr inttoptr(i32 805874800 to ptr), ptr @goo
  ret void
}
```

```
; Check that for i16 type, the maximum legal offset is 62.
; CHECK: foo_i16
; CHECK-DAG: %[[C1:const[0-9]?]] = bitcast i32 805874752 to i32
; CHECK-DAG: %[[C2:const[0-9]?]] = bitcast i32 805874688 to i32
; CHECK: %0 = inttoptr i32 %[[C2]] to ptr
; CHECK-NEXT: %1 = load volatile i16, ptr %0, align 2
; CHECK-NEXT: %[[M1:const_mat[0-9]?]] = add i32 %[[C2]], 4
; CHECK-NEXT: %2 = inttoptr i32 %[[M1]] to ptr
; CHECK-NEXT: %3 = load volatile i16, ptr %2, align 2
; CHECK-NEXT: %[[M2:const_mat[0-9]?]] = add
i32 %[[C2]], 32
; CHECK-NEXT: %4 = inttoptr i32 %[[M2]] to ptr
; CHECK-NEXT: %5 = load volatile i16, ptr %4, align 2
; CHECK-NEXT: %[[M3:const_mat[0-9]?]] = add i32 %[[C2]], 62
; CHECK-NEXT: %6 = inttoptr i32 %[[M3]] to ptr
; CHECK-NEXT: %7 = load volatile i16, ptr %6, align 2
; CHECK-NEXT: %8 = inttoptr i32 %[[C1]] to ptr
; CHECK-NEXT: %9 = load volatile i16, ptr %8, align 2
; CHECK-NEXT: %[[M4:const_mat[0-9]?]] = add i32 %[[C1]], 22
; CHECK-NEXT: %10 = inttoptr i32 %[[M4]] to ptr
; CHECK-NEXT: %11 = load volatile i16, ptr %10, align 2
```

```
define void @foo_i16() {
```

```

entry:
%0 = load volatile i16, ptr inttoptr (i32 805874688 to ptr), align 2
%1 = load volatile i16, ptr inttoptr (i32 805874692 to ptr), align 2
%2 = load volatile i16, ptr inttoptr (i32 805874720 to ptr), align 2
%3 = load volatile i16, ptr inttoptr (i32 805874750 to ptr), align 2
%4 = load volatile i16, ptr inttoptr (i32 805874752 to ptr), align 2
%5 = load volatile i16, ptr inttoptr (i32 805874774 to ptr), align
2
ret void
}

; Check that for i32 type, the maximum legal offset is 124.
; CHECK: foo_i32
; CHECK-DAG: %[[C1:const[0-9]?]] = bitcast i32 805874816 to i32
; CHECK-DAG: %[[C2:const[0-9]?]] = bitcast i32 805874688 to i32
; CHECK: %0 = inttoptr i32 %[[C2]] to ptr
; CHECK-NEXT: %1 = load volatile i32, ptr %0, align 4
; CHECK-NEXT: %[[M1:const_mat[0-9]?]] = add i32 %[[C2]], 4
; CHECK-NEXT: %2 = inttoptr i32 %[[M1]] to ptr
; CHECK-NEXT: %3 = load volatile i32, ptr %2, align 4
; CHECK-NEXT: %[[M2:const_mat[0-9]?]] = add i32 %[[C2]], 124
; CHECK-NEXT: %4 = inttoptr i32 %[[M2]] to ptr
; CHECK-NEXT: %5 = load volatile i32, ptr %4, align 4
; CHECK-NEXT: %6 = inttoptr i32 %[[C1]] to ptr
; CHECK-NEXT: %7 = load volatile i32, ptr %6, align 4
; CHECK-NEXT: %[[M3:const_mat[0-9]?]] = add i32 %[[C1]], 8
; CHECK-NEXT: %8 = inttoptr i32 %[[M3]] to ptr
; CHECK-NEXT: %9 = load volatile i32, ptr %8, align 4
; CHECK-NEXT: %[[M4:const_mat[0-9]?]] = add i32 %[[C1]], 12
; CHECK-NEXT: %10 = inttoptr
i32 %[[M4]] to ptr
; CHECK-NEXT: %11 = load volatile i32, ptr %10, align 4

define void @foo_i32() {
entry:
%0 = load volatile i32, ptr inttoptr (i32 805874688 to ptr), align 4
%1 = load volatile i32, ptr inttoptr (i32 805874692 to ptr), align 4
%2 = load volatile i32, ptr inttoptr (i32 805874812 to ptr), align 4
%3 = load volatile i32, ptr inttoptr (i32 805874816 to ptr), align 4
%4 = load volatile i32, ptr inttoptr (i32 805874824 to ptr), align 4
%5 = load volatile i32, ptr inttoptr (i32 805874828 to ptr), align 4
ret void
}

; RUN: llc -march=hexagon < %s
; REQUIRES: asserts

; The two loads based on %struct.0, loading two different data types
; cause LSR to assume type "void" for the memory type. This would then

```



; cause an assert in isLegalAddressingMode. Make sure we no longer crash.

target triple = "hexagon"

%struct.0 = type { ptr, i8, %union.anon.0 }

%union.anon.0 = type { ptr }

define hidden fastcc void @fred() unnamed\_addr #0 {

entry:

br i1 undef, label %while.end, label %while.body.lr.ph

while.body.lr.ph: ; preds = %entry

br label %while.body

while.body: ; preds = %exit.2, %while.body.lr.ph

%lsr.iv = phi ptr [ %cgep22, %exit.2 ], [ undef, %while.body.lr.ph ]

switch i32 undef, label %exit [

i32 1, label %sw.bb.i

i32 2, label %sw.bb3.i

]

sw.bb.i: ; preds = %while.body

unreachable

sw.bb3.i: ; preds = %while.body

unreachable

exit:

; preds = %while.body

switch i32 undef, label %exit.2 [

i32 1, label %sw.bb.i17

i32 2, label %sw.bb3.i20

]

sw.bb.i17: ; preds = %exit

%0 = load i32, ptr %lsr.iv, align 4

unreachable

sw.bb3.i20: ; preds = %exit

%1 = load ptr, ptr %lsr.iv, align 4

unreachable

exit.2: ; preds = %exit

%cgep22 = getelementptr %struct.0, ptr %lsr.iv, i32 1

br label %while.body

while.end: ; preds = %entry

ret void

```
}
```

```
attributes #0 = { nounwind optsize "target-cpu"="hexagonv55" }
```

```
; NOTE: Assertions have been autogenerated by utils/update_test_checks.py UTC_ARGS: --include-generated-funcs
```

```
; RUN: opt -S -passes=verify,iroutliner -ir-outlining-no-cost < %s | FileCheck %s
```

```
; FIXME: Disabled run-line after running into some turbulence with debuginfo-iterators
```

```
; run: opt -S -passes=verify,iroutliner -ir-outlining-no-cost < %s --try-experimental-debuginfo-iterators | FileCheck %s
```

```
; This test checks that debug info is recognized as able to be extracted along
```

```
; with the other instructions, but is not included in the consolidated function.
```

```
define void @function1() !dbg !6 {
```

```
entry:
```

```
  %a = alloca i32, align 4, !dbg !17
```

```
  call void @llvm.dbg.value(metadata ptr %a, metadata !9, metadata !DIExpression()), !dbg !17
```

```
  %b = alloca i32, align 4, !dbg !18
```

```
  call void @llvm.dbg.value(metadata ptr %b, metadata !11, metadata !DIExpression()), !dbg !18
```

```
  %c = alloca i32, align 4, !dbg !19
```

```
  call void @llvm.dbg.value(metadata ptr %c, metadata !12, metadata !DIExpression()), !dbg !19
```

```
  store i32 2, ptr %a, align 4, !dbg !20
```

```
  store i32 3, ptr %b, align 4, !dbg !21
```

```
  store i32 4, ptr %c, align 4, !dbg !22
```

```
  %al = load i32, ptr %a, align 4, !dbg !23
```

```
  call void @llvm.dbg.value(metadata i32 %al, metadata !13, metadata !DIExpression()), !dbg !23
```

```
  %bl = load i32, ptr %b, align 4, !dbg !24
```

```
  call void @llvm.dbg.value(metadata i32 %bl, metadata !15, metadata !DIExpression()), !dbg !24
```

```
  %cl = load i32, ptr %c, align 4, !dbg !25
```

```
  call void @llvm.dbg.value(metadata i32 %cl, metadata !16, metadata !DIExpression()), !dbg !25
```

```
  ret void, !dbg !26
```

```
}
```

```
define void @function2() !dbg !27 {
```

```
entry:
```

```
  %a = alloca i32, align 4, !dbg !35
```

```
  call void @llvm.dbg.value(metadata ptr %a, metadata !29, metadata !DIExpression()), !dbg !35
```

```
  %b = alloca i32, align 4, !dbg !36
```

```
  call void @llvm.dbg.value(metadata ptr %b, metadata !30, metadata !DIExpression()), !dbg !36
```

```
  %c = alloca i32, align 4, !dbg !37
```

```
  call void @llvm.dbg.value(metadata ptr %c, metadata !31, metadata !DIExpression()), !dbg !37
```

```
  store i32 2, ptr %a, align 4, !dbg !38
```

```
  store i32 3, ptr %b, align 4, !dbg !39
```

```
  store i32 4, ptr %c, align 4, !dbg !40
```

```
  %al = load i32, ptr %a, align 4, !dbg !41
```

```
  call void @llvm.dbg.value(metadata i32 %al, metadata !32, metadata !DIExpression()), !dbg !41
```

```

%bl = load i32, ptr %b, align 4, !dbg !42
call void @llvm.dbg.value(metadata i32 %bl, metadata !33, metadata !DIExpression()), !dbg !42
%cl = load i32, ptr %c, align 4, !dbg !43
call void @llvm.dbg.value(metadata i32 %cl, metadata !34, metadata !DIExpression()), !dbg !43
ret void, !dbg !44
}

```

```

; Function Attrs: nounwind readnone speculatable willreturn
declare void @llvm.dbg.value(metadata, metadata, metadata) #0

```

```

attributes #0 = { nounwind readnone speculatable willreturn }

```

```

!llvm.dbg.cu = !{!0}
!llvm.debugify = !{!3, !4}
!llvm.module.flags = !{!5}

```

```

!0 = distinct !DICompileUnit(language: DW_LANG_C, file: !1, producer: "debugify", isOptimized: true,
runtimeVersion: 0, emissionKind: FullDebug, enums: !2)
!1 = !DIFile(filename:
"legal-debug.ll", directory: "/")
!2 = !{}
!3 = !{i32 20}
!4 = !{i32 12}
!5 = !{i32 2, !"Debug Info Version", i32 3}
!6 = distinct !DISubprogram(name: "function1", linkageName: "function1", scope: null, file: !1, line: 1, type: !7,
scopeLine: 1, spFlags: DISPFlagDefinition | DISPFlagOptimized, unit: !0, retainedNodes: !8)
!7 = !DISubroutineType(types: !2)
!8 = !{!9, !11, !12, !13, !15, !16}
!9 = !DILocalVariable(name: "1", scope: !6, file: !1, line: 1, type: !10)
!10 = !DIBasicType(name: "ty64", size: 64, encoding: DW_ATE_unsigned)
!11 = !DILocalVariable(name: "2", scope: !6, file: !1, line: 2, type: !10)
!12 = !DILocalVariable(name: "3", scope: !6, file: !1, line: 3, type: !10)
!13 = !DILocalVariable(name: "4", scope: !6, file: !1, line: 7, type: !14)
!14 = !DIBasicType(name: "ty32", size: 32, encoding: DW_ATE_unsigned)
!15 = !DILocalVariable(name: "5", scope: !6, file: !1, line: 8, type: !14)
!16 = !DILocalVariable(name: "6", scope: !6, file: !1, line: 9, type: !14)
!17
= !DILocation(line: 1, column: 1, scope: !6)
!18 = !DILocation(line: 2, column: 1, scope: !6)
!19 = !DILocation(line: 3, column: 1, scope: !6)
!20 = !DILocation(line: 4, column: 1, scope: !6)
!21 = !DILocation(line: 5, column: 1, scope: !6)
!22 = !DILocation(line: 6, column: 1, scope: !6)
!23 = !DILocation(line: 7, column: 1, scope: !6)
!24 = !DILocation(line: 8, column: 1, scope: !6)
!25 = !DILocation(line: 9, column: 1, scope: !6)
!26 = !DILocation(line: 10, column: 1, scope: !6)

```

```

!27 = distinct !DISubprogram(name: "function2", linkageName: "function2", scope: null, file: !1, line: 11, type: !7,
scopeLine: 11, spFlags: DISPFlagDefinition | DISPFlagOptimized, unit: !0, retainedNodes: !28)
!28 = !{!29, !30, !31, !32, !33, !34}
!29 = !DILocalVariable(name: "7", scope: !27, file: !1, line: 11, type: !10)
!30 = !DILocalVariable(name: "8", scope: !27, file: !1, line: 12, type: !10)
!31 = !DILocalVariable(name: "9", scope: !27, file: !1, line: 13, type: !10)
!32 = !DILocalVariable(name: "10",
scope: !27, file: !1, line: 17, type: !14)
!33 = !DILocalVariable(name: "11", scope: !27, file: !1, line: 18, type: !14)
!34 = !DILocalVariable(name: "12", scope: !27, file: !1, line: 19, type: !14)
!35 = !DILocation(line: 11, column: 1, scope: !27)
!36 = !DILocation(line: 12, column: 1, scope: !27)
!37 = !DILocation(line: 13, column: 1, scope: !27)
!38 = !DILocation(line: 14, column: 1, scope: !27)
!39 = !DILocation(line: 15, column: 1, scope: !27)
!40 = !DILocation(line: 16, column: 1, scope: !27)
!41 = !DILocation(line: 17, column: 1, scope: !27)
!42 = !DILocation(line: 18, column: 1, scope: !27)
!43 = !DILocation(line: 19, column: 1, scope: !27)
!44 = !DILocation(line: 20, column: 1, scope: !27)
; CHECK-LABEL: @function1(
; CHECK-NEXT: entry:
; CHECK-NEXT: [[A:%.*]] = alloca i32, align 4, !dbg [[DBG17:![0-9]+]]
; CHECK-NEXT: call void @llvm.dbg.value(metadata ptr [[A]], metadata [[META9:![0-9]+]], metadata
!DIExpression()), !dbg [[DBG17]]
; CHECK-NEXT: [[B:%.*]] = alloca
i32, align 4, !dbg [[DBG18:![0-9]+]]
; CHECK-NEXT: call void @llvm.dbg.value(metadata ptr [[B]], metadata [[META11:![0-9]+]], metadata
!DIExpression()), !dbg [[DBG18]]
; CHECK-NEXT: [[C:%.*]] = alloca i32, align 4, !dbg [[DBG19:![0-9]+]]
; CHECK-NEXT: call void @llvm.dbg.value(metadata ptr [[C]], metadata [[META12:![0-9]+]], metadata
!DIExpression()), !dbg [[DBG19]]
; CHECK-NEXT: call void @outlined_ir_func_0(ptr [[A]], ptr [[B]], ptr [[C]]), !dbg [[DBG20:![0-9]+]]
; CHECK-NEXT: ret void, !dbg [[DBG21:![0-9]+]]
;
;
; CHECK-LABEL: @function2(
; CHECK-NEXT: entry:
; CHECK-NEXT: [[A:%.*]] = alloca i32, align 4, !dbg [[DBG30:![0-9]+]]
; CHECK-NEXT: call void @llvm.dbg.value(metadata ptr [[A]], metadata [[META24:![0-9]+]], metadata
!DIExpression()), !dbg [[DBG30]]
; CHECK-NEXT: [[B:%.*]] = alloca i32, align 4, !dbg [[DBG31:![0-9]+]]
; CHECK-NEXT: call void @llvm.dbg.value(metadata ptr [[B]], metadata [[META25:![0-9]+]], metadata
!DIExpression()), !dbg [[DBG31]]
;
CHECK-NEXT: [[C:%.*]] = alloca i32, align 4, !dbg [[DBG32:![0-9]+]]
; CHECK-NEXT: call void @llvm.dbg.value(metadata ptr [[C]], metadata [[META26:![0-9]+]], metadata
!DIExpression()), !dbg [[DBG32]]

```

```
; CHECK-NEXT: call void @outlined_ir_func_0(ptr [[A]], ptr [[B]], ptr [[C]]), !dbg [[DBG33:![0-9]+]]
; CHECK-NEXT: ret void, !dbg [[DBG34:![0-9]+]]
;
;
; CHECK: @outlined_ir_func_0(ptr [[TMP0:%.*]], ptr [[TMP1:%.*]], ptr [[TMP2:%.*]])
; CHECK: entry_to_outline:
; CHECK-NEXT: store i32 2, ptr [[TMP0]], align 4
; CHECK-NEXT: store i32 3, ptr [[TMP1]], align 4
; CHECK-NEXT: store i32 4, ptr [[TMP2]], align 4
; CHECK-NEXT: [[AL:%.*]] = load i32, ptr [[TMP0]], align 4
; CHECK-NEXT: [[BL:%.*]] = load i32, ptr [[TMP1]], align 4
; CHECK-NEXT: [[CL:%.*]] = load i32, ptr [[TMP2]], align 4
; CHECK-NEXT: br label [[ENTRY_AFTER_OUTLINE_EXITSTUB:%.*]]
;
; RUN: opt < %s -passes=argpromotion -S | FileCheck %s
```

```
; CHECK-LABEL: define i32 @foo() #0 {
; CHECK-NEXT:    %.val = load <32 x half>, ptr undef, align 4
; CHECK-NEXT:    call void @bar(<32 x half> %.val)
; CHECK-NEXT:    ret i32 0
; CHECK-NEXT: }

; CHECK-LABEL: define internal void @bar(<32 x half> %.0.val) #0 {
; CHECK-NEXT:    ret void
; CHECK-NEXT: }
```

```
; CHECK: attributes #0 = { uwtable "min-legal-vector-width"="512" }
```

```
define i32 @foo() #0 {
  call void @bar(ptr undef)
  ret i32 0
}
```

```
define internal void @bar(ptr) #0 {
  %2 = load <32 x half>, ptr %0, align 4
  ret void
}
```

```
attributes #0 = { uwtable "min-legal-vector-width"="0" }
```

```
=====
The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:
=====
```

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

## 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the

power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner

or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition,

"submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices

stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.



7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions  
of this Software are embedded into an Object form of such source code, you  
may redistribute such embedded portions in such Object form without complying  
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled  
forms of this Software with  
software that is licensed under the GPLv2 ("Combined Software") and if a  
court of competent jurisdiction determines that the patent provision (Section  
3), the indemnity provision (Section 9) or other Section of the License  
conflicts with the conditions of the GPLv2, you may retroactively and  
prospectively choose to deem waived or otherwise exclude such Section(s) of  
the License, but only in their entirety and only with respect to the Combined  
Software.

=====  
Software from third parties included in the LLVM Project:  
=====

The LLVM Project contains third party software which is under different license  
terms. All such code will be identified clearly using at least one of two  
mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or  
`LICENSE` file at the top containing the specific  
license and restrictions  
which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every  
file.

=====  
Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):  
=====

The libc++ library is dual licensed under both the University of Illinois "BSD-Like" license and the MIT license. As a user of this code you may choose to use it under either license. As a contributor, you agree to allow your code to be used under both.

Full text of the relevant licenses is included below.

=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2009-2019 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

<http://llvm.org>

#### Permission

is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- \* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS

FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====

Copyright (c) 2009-2014 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This work is released into the public domain with CC0 1.0. Alternatively, it is licensed under the Apache License 2.0.

---

## Creative Commons Legal Code

### CC0 1.0 Universal

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS DOCUMENT DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED HEREUNDER, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED HEREUNDER.

### Statement of Purpose

The laws of most jurisdictions throughout the world automatically confer exclusive Copyright and Related Rights (defined below) upon the creator and subsequent owner(s) (each and all, an "owner") of an original work of authorship and/or a database (each, a "Work").

Certain owners wish to permanently relinquish those rights to a Work for the purpose of contributing to a commons of creative, cultural and scientific works ("Commons") that the public can reliably and without fear of later claims of infringement build upon, modify, incorporate in other works, reuse and redistribute as freely as possible in any form whatsoever and for any purposes, including without limitation commercial purposes. These owners may contribute to the Commons to promote the ideal of a free culture and the further production of creative, cultural and scientific works, or to gain reputation or greater distribution for their Work in part through the use and efforts of others.

For these and/or other purposes and motivations, and without any expectation of additional consideration or compensation, the person associating CC0 with a Work (the "Affirmer"), to the extent that he or she is an owner of Copyright and Related Rights in the Work, voluntarily elects to apply CC0 to the Work and publicly distribute the Work under its terms, with knowledge of his or her Copyright and Related Rights in the Work and the meaning and intended legal effect of CC0 on those rights.

1. Copyright and Related Rights. A Work made available under CC0 may be protected by copyright and related or neighboring rights ("Copyright and

Related Rights"). Copyright and Related Rights include, but are not limited to, the following:

- i. the right to reproduce, adapt, distribute, perform, display, communicate, and translate a Work;
- ii. moral rights retained by the original author(s) and/or performer(s);
- iii. publicity and privacy rights pertaining to a person's image or likeness depicted in a Work;
- iv. rights protecting against unfair competition in regards to a Work, subject to the limitations in paragraph 4(a), below;
- v. rights protecting the extraction, dissemination, use and reuse of data in a Work;
- vi. database rights (such as those arising under Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, and under any national implementation thereof, including any amended or successor version of such directive); and
- vii. other similar, equivalent or corresponding rights throughout the world based on applicable law or treaty, and any national implementations thereof.

2. Waiver. To the greatest extent permitted by, but not in contravention of, applicable law, Affirmer hereby overtly, fully, permanently, irrevocably and unconditionally waives, abandons, and surrenders all of Affirmer's Copyright and Related Rights and associated claims and causes of action, whether now known or unknown (including existing as well as future claims and causes of action), in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "Waiver"). Affirmer makes the Waiver for the benefit of each member of the public at large and to the detriment of Affirmer's heirs and successors, fully intending that such Waiver shall not be subject to revocation, rescission, cancellation, termination, or any other legal or equitable action to disrupt the quiet enjoyment of the Work by the public as contemplated by Affirmer's express Statement of Purpose.

3. Public License Fallback. Should any part of the Waiver for any reason be judged legally invalid or ineffective under applicable law, then the Waiver shall be preserved to the maximum extent permitted taking into account Affirmer's express Statement of Purpose. In addition, to the extent the Waiver is so judged Affirmer hereby grants to each affected person a royalty-free, non transferable, non sublicensable, non exclusive, irrevocable and unconditional license to exercise Affirmer's Copyright and

Related Rights in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "License"). The License shall be deemed effective as of the date CC0 was applied by Affirmer to the Work. Should any part of the License for any reason be judged legally invalid or ineffective under applicable law, such partial invalidity or ineffectiveness shall not invalidate the remainder of the License, and in such case Affirmer hereby affirms that he or she will not (i) exercise any of his or her remaining Copyright and Related Rights in the Work or (ii) assert any associated claims and causes of action with respect to the Work, in either case contrary to Affirmer's express Statement of Purpose.

#### 4. Limitations and Disclaimers.

- a. No trademark or patent rights held by Affirmer are waived, abandoned, surrendered, licensed or otherwise affected by this document.
- b. Affirmer offers the Work as-is and makes no representations or warranties of any kind concerning the Work, express, implied, statutory or otherwise, including without limitation warranties of title, merchantability, fitness for a particular purpose, non infringement, or the absence of latent or other defects, accuracy, or the present or absence of errors, whether or not discoverable, all to the greatest extent permissible under applicable law.
- c. Affirmer disclaims responsibility for clearing rights of other persons that may apply to the Work or any use thereof, including without limitation any person's Copyright and Related Rights in the Work. Further, Affirmer disclaims responsibility for obtaining any necessary consents, permissions or other rights required for any use of the Work.
- d. Affirmer understands and acknowledges that Creative Commons is not a party to this document and has no duty or obligation with respect to this CC0 or use of the Work.

-----

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent



to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability

obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

## APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page"

as the copyright notice for easier

identification within third-party archives.

Copyright 2019 Jack O'Connor and Samuel Neves

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -mattr=+m -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefixes=CHECK,RV64I
; RUN: llc -mtriple=riscv64 -mattr=+m,+zba -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-
prefixes=CHECK,RV64ZBA,RV64ZBANOZBB
; RUN: llc -mtriple=riscv64 -mattr=+m,+zba,+zbb -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefixes=CHECK,RV64ZBA,RV64ZBAZBB
```

```
define i64 @slliuw(i64 %a) nounwind {
; RV64I-LABEL: slliuw:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 31
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: slliuw:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  slli.uw a0, a0, 1
; RV64ZBA-NEXT:  ret
%conv1 = shl i64 %a, 1
%shl = and i64 %conv1, 8589934590
ret i64 %shl
}
```

```
define i128 @slliuw_2(i32 signext %0, ptr %1) {
; RV64I-LABEL:
slliuw_2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 28
; RV64I-NEXT:  add a1, a1, a0
; RV64I-NEXT:  ld a0, 0(a1)
; RV64I-NEXT:  ld a1, 8(a1)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: slliuw_2:
; RV64ZBA:      # %bb.0:
```

```

; RV64ZBA-NEXT:  slli.uw a0, a0, 4
; RV64ZBA-NEXT:  add a1, a1, a0
; RV64ZBA-NEXT:  ld a0, 0(a1)
; RV64ZBA-NEXT:  ld a1, 8(a1)
; RV64ZBA-NEXT:  ret
%3 = zext i32 %0 to i64
%4 = getelementptr inbounds i128, ptr %1, i64 %3
%5 = load i128, ptr %4
ret i128 %5
}

```

```

define i128 @slliuv_3(i32 signext %0, ptr %1) {
; RV64I-LABEL: slliuv_3:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi a0, a0, 1
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 28
; RV64I-NEXT:  add a1, a1, a0
; RV64I-NEXT:  ld a0, 0(a1)
; RV64I-NEXT:  ld a1, 8(a1)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: slliuv_3:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  addi a0, a0, 1
; RV64ZBA-NEXT:  slli.uw a0, a0, 4
; RV64ZBA-NEXT:
    add a1, a1, a0
; RV64ZBA-NEXT:  ld a0, 0(a1)
; RV64ZBA-NEXT:  ld a1, 8(a1)
; RV64ZBA-NEXT:  ret
%add = add i32 %0, 1
%3 = zext i32 %add to i64
%4 = getelementptr inbounds i128, ptr %1, i64 %3
%5 = load i128, ptr %4
ret i128 %5
}

```

```

define i64 @adduw(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: adduw:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  srli a1, a1, 32
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: adduw:
; RV64ZBA:      # %bb.0:

```

```

; RV64ZBA-NEXT:  add.uw a0, a1, a0
; RV64ZBA-NEXT:  ret
%and = and i64 %b, 4294967295
%add = add i64 %and, %a
ret i64 %add
}

define signext i8 @adduw_2(i32 signext %0, ptr %1) {
; RV64I-LABEL: adduw_2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  lb a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: adduw_2:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  add.uw a0, a0, a1
; RV64ZBA-NEXT:
    lb a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = zext i32 %0 to i64
%4 = getelementptr inbounds i8, ptr %1, i64 %3
%5 = load i8, ptr %4
ret i8 %5
}

define signext i8 @adduw_3(i32 signext %0, ptr %1) {
; RV64I-LABEL: adduw_3:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi a0, a0, 1
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  lb a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: adduw_3:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  addi a0, a0, 1
; RV64ZBA-NEXT:  add.uw a0, a0, a1
; RV64ZBA-NEXT:  lb a0, 0(a0)
; RV64ZBA-NEXT:  ret
%add = add i32 %0, 1
%3 = zext i32 %add to i64
%4 = getelementptr inbounds i8, ptr %1, i64 %3
%5 = load i8, ptr %4

```

```
ret i8 %5
}
```

```
define i64 @zextw_i64(i64 %a) nounwind {
; RV64I-LABEL: zextw_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: zextw_i64:
; RV64ZBA:     # %bb.0:
; RV64ZBA-NEXT:
    zext.w a0, a0
; RV64ZBA-NEXT:  ret
%and = and i64 %a, 4294967295
ret i64 %and
}
```

; This makes sure targetShrinkDemandedConstant changes the and immediate to  
; allow zext.w or slli+srli.

```
define i64 @zextw_demandedbits_i64(i64 %0) {
; RV64I-LABEL: zextw_demandedbits_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  ori a0, a0, 1
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: zextw_demandedbits_i64:
; RV64ZBA:     # %bb.0:
; RV64ZBA-NEXT:  ori a0, a0, 1
; RV64ZBA-NEXT:  zext.w a0, a0
; RV64ZBA-NEXT:  ret
%2 = and i64 %0, 4294967294
%3 = or i64 %2, 1
ret i64 %3
}
```

```
define signext i16 @sh1add(i64 %0, ptr %1) {
; RV64I-LABEL: sh1add:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 1
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  lh a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh1add:
```

```
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh1add a0, a0, a1
; RV64ZBA-NEXT:  lh a0, 0(a0)
; RV64ZBA-NEXT:
    ret
%3 = getelementptr inbounds i16, ptr %1, i64 %0
%4 = load i16, ptr %3
ret i16 %4
}
```

```
define signext i32 @sh2add(i64 %0, ptr %1) {
; RV64I-LABEL: sh2add:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 2
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  lw a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh2add:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh2add a0, a0, a1
; RV64ZBA-NEXT:  lw a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = getelementptr inbounds i32, ptr %1, i64 %0
%4 = load i32, ptr %3
ret i32 %4
}
```

```
define i64 @sh3add(i64 %0, ptr %1) {
; RV64I-LABEL: sh3add:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 3
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  ld a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh3add:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh3add a0, a0, a1
; RV64ZBA-NEXT:  ld a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = getelementptr inbounds i64, ptr %1, i64 %0
%4 = load i64, ptr %3
ret i64 %4
}
```

```
define signext i16 @sh1adduw(i32
```



```

signext %0, ptr %1) {
; RV64I-LABEL: sh1adduw:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 31
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  lh a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh1adduw:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh1add.uw a0, a0, a1
; RV64ZBA-NEXT:  lh a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = zext i32 %0 to i64
%4 = getelementptr inbounds i16, ptr %1, i64 %3
%5 = load i16, ptr %4
ret i16 %5
}

define i64 @sh1adduw_2(i64 %0, i64 %1) {
; RV64I-LABEL: sh1adduw_2:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 31
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh1adduw_2:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh1add.uw a0, a0, a1
; RV64ZBA-NEXT:  ret
%3 = shl i64 %0, 1
%4 = and i64 %3, 8589934590
%5 = add i64 %4, %1
ret i64 %5
}

define signext i32 @sh2adduw(i32 signext %0, ptr %1) {
; RV64I-LABEL: sh2adduw:
; RV64I:
;   # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 30
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  lw a0, 0(a0)
; RV64I-NEXT:  ret
;

```

```

; RV64ZBA-LABEL: sh2adduw:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh2add.uw a0, a0, a1
; RV64ZBA-NEXT:  lw a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = zext i32 %0 to i64
%4 = getelementptr inbounds i32, ptr %1, i64 %3
%5 = load i32, ptr %4
ret i32 %5
}

```

```

define i64 @sh2adduw_2(i64 %0, i64 %1) {
; RV64I-LABEL: sh2adduw_2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 30
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh2adduw_2:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh2add.uw a0, a0, a1
; RV64ZBA-NEXT:  ret
%3 = shl i64 %0, 2
%4 = and i64 %3, 17179869180
%5 = add i64 %4, %1
ret i64 %5
}

```

```

define i64 @sh3adduw(i32 signext %0, ptr %1) {
; RV64I-LABEL: sh3adduw:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:
; RV64I-NEXT:  srli a0, a0, 29
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  ld a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh3adduw:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh3add.uw a0, a0, a1
; RV64ZBA-NEXT:  ld a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = zext i32 %0 to i64
%4 = getelementptr inbounds i64, ptr %1, i64 %3
%5 = load i64, ptr %4
ret i64 %5

```

```
}
```

```
define i64 @sh3adduw_2(i64 %0, i64 %1) {  
; RV64I-LABEL: sh3adduw_2:  
; RV64I:      # %bb.0:  
; RV64I-NEXT: slli a0, a0, 32  
; RV64I-NEXT: srli a0, a0, 29  
; RV64I-NEXT: add a0, a0, a1  
; RV64I-NEXT: ret  
;  
; RV64ZBA-LABEL: sh3adduw_2:  
; RV64ZBA:    # %bb.0:  
; RV64ZBA-NEXT: sh3add.uw a0, a0, a1  
; RV64ZBA-NEXT: ret  
%3 = shl i64 %0, 3  
%4 = and i64 %3, 34359738360  
%5 = add i64 %4, %1  
ret i64 %5  
}
```

; Type legalization inserts a sext\_inreg after the first add. That add will be  
; selected as sh2add which does not sign extend. SimplifyDemandedBits is unable  
; to remove the sext\_inreg because it has multiple uses.

The ashr will use the

; sext\_inreg to become srai. This leaves the sext\_inreg only used by the shl.

; If the shl is selected as sllw, we don't need the sext\_inreg.

```
define i64 @sh2add_extra_sext(i32 %x, i32 %y, i32 %z) {
```

```
; RV64I-LABEL: sh2add_extra_sext:  
; RV64I:      # %bb.0:  
; RV64I-NEXT: slli a0, a0, 2  
; RV64I-NEXT: add a0, a0, a1  
; RV64I-NEXT: sllw a1, a2, a0  
; RV64I-NEXT: srai a0, a0, 2  
; RV64I-NEXT: mul a0, a1, a0  
; RV64I-NEXT: ret  
;  
; RV64ZBA-LABEL: sh2add_extra_sext:  
; RV64ZBA:    # %bb.0:  
; RV64ZBA-NEXT: sh2add a0, a0, a1  
; RV64ZBA-NEXT: sllw a1, a2, a0  
; RV64ZBA-NEXT: srai a0, a0, 2  
; RV64ZBA-NEXT: mul a0, a1, a0  
; RV64ZBA-NEXT: ret  
%a = shl i32 %x, 2  
%b = add i32 %a, %y  
%c = shl i32 %z, %b  
%d = ashr i32 %b, 2  
%e = sext i32 %c to i64
```

```

%f = sext i32 %d to i64
%g = mul i64 %e, %f
ret i64 %g
}

```

```

define i64 @addmul6(i64 %a, i64 %b) {
; RV64I-LABEL: addmul6:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a2, 6
; RV64I-NEXT:  mul
a0, a0, a2
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addmul6:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT: sh1add a0, a0, a0
; RV64ZBA-NEXT: sh1add a0, a0, a1
; RV64ZBA-NEXT: ret
%c = mul i64 %a, 6
%d = add i64 %c, %b
ret i64 %d
}

```

```

define i64 @addmul10(i64 %a, i64 %b) {
; RV64I-LABEL: addmul10:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a2, 10
; RV64I-NEXT:  mul a0, a0, a2
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addmul10:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT: sh2add a0, a0, a0
; RV64ZBA-NEXT: sh1add a0, a0, a1
; RV64ZBA-NEXT: ret
%c = mul i64 %a, 10
%d = add i64 %c, %b
ret i64 %d
}

```

```

define i64 @addmul12(i64 %a, i64 %b) {
; RV64I-LABEL: addmul12:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a2, 12
; RV64I-NEXT:  mul a0, a0, a2
; RV64I-NEXT:  add a0, a0, a1

```

```

; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addmul12:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh1add a0, a0, a0
; RV64ZBA-NEXT:  sh2add
a0, a0, a1
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 12
%d = add i64 %c, %b
ret i64 %d
}

```

```

define i64 @addmul18(i64 %a, i64 %b) {
; RV64I-LABEL: addmul18:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a2, 18
; RV64I-NEXT:  mul a0, a0, a2
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addmul18:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh3add a0, a0, a0
; RV64ZBA-NEXT:  sh1add a0, a0, a1
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 18
%d = add i64 %c, %b
ret i64 %d
}

```

```

define i64 @addmul20(i64 %a, i64 %b) {
; RV64I-LABEL: addmul20:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a2, 20
; RV64I-NEXT:  mul a0, a0, a2
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addmul20:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh2add a0, a0, a0
; RV64ZBA-NEXT:  sh2add a0, a0, a1
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 20
%d = add i64 %c, %b
ret i64 %d
}

```

```

define i64 @addmul24(i64 %a, i64 %b) {
; RV64I-LABEL: addmul24:
; RV64I:
;   # %bb.0:
; RV64I-NEXT:  li a2, 24
; RV64I-NEXT:  mul a0, a0, a2
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addmul24:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh1add a0, a0, a0
; RV64ZBA-NEXT:  sh3add a0, a0, a1
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 24
%d = add i64 %c, %b
ret i64 %d
}

```

```

define i64 @addmul36(i64 %a, i64 %b) {
; RV64I-LABEL: addmul36:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a2, 36
; RV64I-NEXT:  mul a0, a0, a2
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addmul36:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh3add a0, a0, a0
; RV64ZBA-NEXT:  sh2add a0, a0, a1
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 36
%d = add i64 %c, %b
ret i64 %d
}

```

```

define i64 @addmul40(i64 %a, i64 %b) {
; RV64I-LABEL: addmul40:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a2, 40
; RV64I-NEXT:  mul a0, a0, a2
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addmul40:
; RV64ZBA:    # %bb.0:

```

```
;
RV64ZBA-NEXT: sh2add a0, a0, a0
; RV64ZBA-NEXT: sh3add a0, a0, a1
; RV64ZBA-NEXT: ret
%c = mul i64 %a, 40
%d = add i64 %c, %b
ret i64 %d
}
```

```
define i64 @addmul72(i64 %a, i64 %b) {
; RV64I-LABEL: addmul72:
; RV64I:      # %bb.0:
; RV64I-NEXT: li a2, 72
; RV64I-NEXT: mul a0, a0, a2
; RV64I-NEXT: add a0, a0, a1
; RV64I-NEXT: ret
;
; RV64ZBA-LABEL: addmul72:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT: sh3add a0, a0, a0
; RV64ZBA-NEXT: sh3add a0, a0, a1
; RV64ZBA-NEXT: ret
%c = mul i64 %a, 72
%d = add i64 %c, %b
ret i64 %d
}
```

```
define i64 @mul96(i64 %a) {
; RV64I-LABEL: mul96:
; RV64I:      # %bb.0:
; RV64I-NEXT: li a1, 96
; RV64I-NEXT: mul a0, a0, a1
; RV64I-NEXT: ret
;
; RV64ZBA-LABEL: mul96:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT: sh1add a0, a0, a0
; RV64ZBA-NEXT: slli a0, a0, 5
; RV64ZBA-NEXT: ret
%c = mul i64 %a, 96
ret i64 %c
}
```

```
define i64 @mul160(i64 %a) {
; RV64I-LABEL: mul160:
; RV64I:      # %bb.0:
; RV64I-NEXT:
```

```

    li a1, 160
; RV64I-NEXT:    mul a0, a0, a1
; RV64I-NEXT:    ret
;
; RV64ZBA-LABEL: mul160:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh2add a0, a0, a0
; RV64ZBA-NEXT:  slli a0, a0, 5
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 160
ret i64 %c
}

define i64 @mul288(i64 %a) {
; RV64I-LABEL: mul288:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 288
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul288:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh3add a0, a0, a0
; RV64ZBA-NEXT:  slli a0, a0, 5
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 288
ret i64 %c
}

define i64 @zext_mul96(i32 signext %a) {
; RV64I-LABEL: zext_mul96:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 3
; RV64I-NEXT:  slli a1, a1, 37
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  mulhu a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: zext_mul96:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  slli.uw a0, a0, 5
; RV64ZBA-NEXT:  sh1add a0, a0, a0
; RV64ZBA-NEXT:  ret
%b = zext
i32 %a to i64
%c = mul i64 %b, 96
ret i64 %c
}

```



```

define i64 @zext_mul160(i32 signext %a) {
; RV64I-LABEL: zext_mul160:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 5
; RV64I-NEXT:  slli a1, a1, 37
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  mulhu a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: zext_mul160:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  slli.uw a0, a0, 5
; RV64ZBA-NEXT:  sh2add a0, a0, a0
; RV64ZBA-NEXT:  ret
%b = zext i32 %a to i64
%c = mul i64 %b, 160
ret i64 %c
}

```

```

define i64 @zext_mul288(i32 signext %a) {
; RV64I-LABEL: zext_mul288:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 9
; RV64I-NEXT:  slli a1, a1, 37
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  mulhu a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: zext_mul288:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  slli.uw a0, a0, 5
; RV64ZBA-NEXT:  sh3add a0, a0, a0
; RV64ZBA-NEXT:  ret
%b = zext i32 %a to i64
%c = mul i64 %b, 288
ret i64 %c
}

```

; We can't use slli.uw becaues  
the shift amount is more than 31.  
; FIXME: The zext.w is unneeded.

```

define i64 @zext_mul12884901888(i32 signext %a) {
; RV64I-LABEL: zext_mul12884901888:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  li a1, 3

```

```

; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: zext_mul12884901888:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  andi a0, a0, -1
; RV64ZBA-NEXT:  sh1add a0, a0, a0
; RV64ZBA-NEXT:  slli a0, a0, 32
; RV64ZBA-NEXT:  ret
%b = zext i32 %a to i64
%c = mul i64 %b, 12884901888
ret i64 %c
}

```

; We can't use slli.uw becaues the shift amount is more than 31.

; FIXME: The zext.w is unneeded.

```
define i64 @zext_mul21474836480(i32 signext %a) {
```

```
; RV64I-LABEL: zext_mul21474836480:
```

```
; RV64I:      # %bb.0:
```

```
; RV64I-NEXT:  slli a0, a0, 32
```

```
; RV64I-NEXT:  srli a0, a0, 32
```

```
; RV64I-NEXT:  li a1, 5
```

```
; RV64I-NEXT:  slli a1, a1, 32
```

```
; RV64I-NEXT:  mul a0, a0, a1
```

```
; RV64I-NEXT:
```

```
    ret
```

```
;
```

```
; RV64ZBA-LABEL: zext_mul21474836480:
```

```
; RV64ZBA:      # %bb.0:
```

```
; RV64ZBA-NEXT:  andi a0, a0, -1
```

```
; RV64ZBA-NEXT:  sh2add a0, a0, a0
```

```
; RV64ZBA-NEXT:  slli a0, a0, 32
```

```
; RV64ZBA-NEXT:  ret
```

```
%b = zext i32 %a to i64
```

```
%c = mul i64 %b, 21474836480
```

```
ret i64 %c
```

```
}
```

; We can't use slli.uw becaues the shift amount is more than 31.

; FIXME: The zext.w is unneeded.

```
define i64 @zext_mul38654705664(i32 signext %a) {
```

```
; RV64I-LABEL: zext_mul38654705664:
```

```
; RV64I:      # %bb.0:
```

```
; RV64I-NEXT:  slli a0, a0, 32
```

```
; RV64I-NEXT:  srli a0, a0, 32
```

```
; RV64I-NEXT:  li a1, 9
```

```
; RV64I-NEXT:  slli a1, a1, 32
```

```

; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: zext_mul38654705664:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  andi a0, a0, -1
; RV64ZBA-NEXT:  sh3add a0, a0, a0
; RV64ZBA-NEXT:  slli a0, a0, 32
; RV64ZBA-NEXT:  ret
%b = zext i32 %a to i64
%c = mul i64 %b, 38654705664
ret i64 %c
}

```

```

define i64 @sh1add_imm(i64 %0) {
; CHECK-LABEL: sh1add_imm:
;
CHECK:      # %bb.0:
; CHECK-NEXT:  slli a0, a0, 1
; CHECK-NEXT:  addi a0, a0, 5
; CHECK-NEXT:  ret
%a = shl i64 %0, 1
%b = add i64 %a, 5
ret i64 %b
}

```

```

define i64 @sh2add_imm(i64 %0) {
; CHECK-LABEL: sh2add_imm:
; CHECK:      # %bb.0:
; CHECK-NEXT:  slli a0, a0, 2
; CHECK-NEXT:  addi a0, a0, -6
; CHECK-NEXT:  ret
%a = shl i64 %0, 2
%b = add i64 %a, -6
ret i64 %b
}

```

```

define i64 @sh3add_imm(i64 %0) {
; CHECK-LABEL: sh3add_imm:
; CHECK:      # %bb.0:
; CHECK-NEXT:  slli a0, a0, 3
; CHECK-NEXT:  addi a0, a0, 7
; CHECK-NEXT:  ret
%a = shl i64 %0, 3
%b = add i64 %a, 7
ret i64 %b
}

```

```

define i64 @sh1adduw_imm(i32 signext %0) {
; RV64I-LABEL: sh1adduw_imm:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 31
; RV64I-NEXT:  addi a0, a0, 11
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh1adduw_imm:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  slli.uw a0, a0, 1
; RV64ZBA-NEXT:  addi a0, a0, 11
; RV64ZBA-NEXT:  ret
%a =
zext i32 %0 to i64
%b = shl i64 %a, 1
%c = add i64 %b, 11
ret i64 %c
}

```

```

define i64 @sh2adduw_imm(i32 signext %0) {
; RV64I-LABEL: sh2adduw_imm:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 30
; RV64I-NEXT:  addi a0, a0, -12
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh2adduw_imm:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  slli.uw a0, a0, 2
; RV64ZBA-NEXT:  addi a0, a0, -12
; RV64ZBA-NEXT:  ret
%a = zext i32 %0 to i64
%b = shl i64 %a, 2
%c = add i64 %b, -12
ret i64 %c
}

```

```

define i64 @sh3adduw_imm(i32 signext %0) {
; RV64I-LABEL: sh3adduw_imm:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 29
; RV64I-NEXT:  addi a0, a0, 13
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh3adduw_imm:

```

```

; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  slli.uw a0, a0, 3
; RV64ZBA-NEXT:  addi a0, a0, 13
; RV64ZBA-NEXT:  ret
%a = zext i32 %0 to i64
%b = shl i64 %a, 3
%c = add i64 %b, 13
ret i64 %c
}

define i64 @adduw_imm(i32 signext %0) nounwind {
; RV64I-LABEL: adduw_imm:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  addi a0, a0, 5
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: adduw_imm:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  zext.w a0, a0
; RV64ZBA-NEXT:  addi a0, a0, 5
; RV64ZBA-NEXT:  ret
%a = zext i32 %0 to i64
%b = add i64 %a, 5
ret i64 %b
}

define i64 @mul258(i64 %a) {
; CHECK-LABEL: mul258:
; CHECK:    # %bb.0:
; CHECK-NEXT:  li a1, 258
; CHECK-NEXT:  mul a0, a0, a1
; CHECK-NEXT:  ret
%c = mul i64 %a, 258
ret i64 %c
}

define i64 @mul260(i64 %a) {
; CHECK-LABEL: mul260:
; CHECK:    # %bb.0:
; CHECK-NEXT:  li a1, 260
; CHECK-NEXT:  mul a0, a0, a1
; CHECK-NEXT:  ret
%c = mul i64 %a, 260
ret i64 %c
}

```

```

define i64 @mul264(i64 %a) {
; CHECK-LABEL: mul264:
; CHECK:      # %bb.0:
; CHECK-NEXT:  li a1, 264
; CHECK-NEXT:  mul a0, a0, a1
; CHECK-NEXT:  ret
%c = mul i64 %a, 264
ret i64
%c
}

```

```

define i64 @imm_zextw() nounwind {
; RV64I-LABEL: imm_zextw:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a0, 1
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  addi a0, a0, -2
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: imm_zextw:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  li a0, -2
; RV64ZBA-NEXT:  zext.w a0, a0
; RV64ZBA-NEXT:  ret
ret i64 4294967294 ; -2 in 32 bits.
}

```

```

define i64 @mul11(i64 %a) {
; RV64I-LABEL: mul11:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 11
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul11:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh2add a1, a0, a0
; RV64ZBA-NEXT:  sh1add a0, a1, a0
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 11
ret i64 %c
}

```

```

define i64 @mul19(i64 %a) {
; RV64I-LABEL: mul19:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 19

```

```

; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul19:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh3add a1, a0, a0
; RV64ZBA-NEXT:  sh1add a0,
a1, a0
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 19
ret i64 %c
}

```

```

define i64 @mul13(i64 %a) {
; RV64I-LABEL: mul13:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 13
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul13:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh1add a1, a0, a0
; RV64ZBA-NEXT:  sh2add a0, a1, a0
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 13
ret i64 %c
}

```

```

define i64 @mul21(i64 %a) {
; RV64I-LABEL: mul21:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 21
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul21:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh2add a1, a0, a0
; RV64ZBA-NEXT:  sh2add a0, a1, a0
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 21
ret i64 %c
}

```

```

define i64 @mul37(i64 %a) {
; RV64I-LABEL: mul37:
; RV64I:      # %bb.0:

```

```

; RV64I-NEXT:  li a1, 37
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul37:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh3add a1, a0, a0
; RV64ZBA-NEXT:
    sh2add a0, a1, a0
; RV64ZBA-NEXT:  ret
    %c = mul i64 %a, 37
    ret i64 %c
}

```

```

define i64 @mul25(i64 %a) {
; RV64I-LABEL: mul25:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 25
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul25:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh1add a1, a0, a0
; RV64ZBA-NEXT:  sh3add a0, a1, a0
; RV64ZBA-NEXT:  ret
    %c = mul i64 %a, 25
    ret i64 %c
}

```

```

define i64 @mul41(i64 %a) {
; RV64I-LABEL: mul41:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 41
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul41:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh2add a1, a0, a0
; RV64ZBA-NEXT:  sh3add a0, a1, a0
; RV64ZBA-NEXT:  ret
    %c = mul i64 %a, 41
    ret i64 %c
}

```

```

define i64 @mul73(i64 %a) {
; RV64I-LABEL: mul73:

```



```

; RV64I:    # %bb.0:
; RV64I-NEXT:  li a1, 73
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul73:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh3add
a1, a0, a0
; RV64ZBA-NEXT:  sh3add a0, a1, a0
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 73
ret i64 %c
}

```

```

define i64 @mul27(i64 %a) {
; RV64I-LABEL: mul27:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a1, 27
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul27:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh3add a0, a0, a0
; RV64ZBA-NEXT:  sh1add a0, a0, a0
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 27
ret i64 %c
}

```

```

define i64 @mul45(i64 %a) {
; RV64I-LABEL: mul45:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a1, 45
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul45:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  sh3add a0, a0, a0
; RV64ZBA-NEXT:  sh2add a0, a0, a0
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 45
ret i64 %c
}

```

```

define i64 @mul81(i64 %a) {

```

```

; RV64I-LABEL: mul81:
; RV64I:    # %bb.0:
; RV64I-NEXT:  li a1, 81
; RV64I-NEXT:  mul a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul81:
; RV64ZBA:    # %bb.0:
;
RV64ZBA-NEXT:  sh3add a0, a0, a0
; RV64ZBA-NEXT:  sh3add a0, a0, a0
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 81
ret i64 %c
}

```

```

define i64 @mul4098(i64 %a) {
; RV64I-LABEL: mul4098:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a1, a0, 1
; RV64I-NEXT:  slli a0, a0, 12
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul4098:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  slli a1, a0, 12
; RV64ZBA-NEXT:  sh1add a0, a0, a1
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 4098
ret i64 %c
}

```

```

define i64 @mul4100(i64 %a) {
; RV64I-LABEL: mul4100:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a1, a0, 2
; RV64I-NEXT:  slli a0, a0, 12
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: mul4100:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  slli a1, a0, 12
; RV64ZBA-NEXT:  sh2add a0, a0, a1
; RV64ZBA-NEXT:  ret
%c = mul i64 %a, 4100
ret i64 %c
}

```

```
}
```

```
define i64 @mul4104(i64 %a) {  
; RV64I-LABEL: mul4104:  
; RV64I:    # %bb.0:  
; RV64I-NEXT:  
    slli a1, a0, 3  
; RV64I-NEXT:  slli a0, a0, 12  
; RV64I-NEXT:  add a0, a0, a1  
; RV64I-NEXT:  ret  
;  
; RV64ZBA-LABEL: mul4104:  
; RV64ZBA:    # %bb.0:  
; RV64ZBA-NEXT:  slli a1, a0, 12  
; RV64ZBA-NEXT:  sh3add a0, a0, a1  
; RV64ZBA-NEXT:  ret  
    %c = mul i64 %a, 4104  
    ret i64 %c  
}
```

```
define signext i32 @mulw192(i32 signext %a) {  
; CHECK-LABEL: mulw192:  
; CHECK:    # %bb.0:  
; CHECK-NEXT:  li a1, 192  
; CHECK-NEXT:  mulw a0, a0, a1  
; CHECK-NEXT:  ret  
    %c = mul i32 %a, 192  
    ret i32 %c  
}
```

```
define signext i32 @mulw320(i32 signext %a) {  
; CHECK-LABEL: mulw320:  
; CHECK:    # %bb.0:  
; CHECK-NEXT:  li a1, 320  
; CHECK-NEXT:  mulw a0, a0, a1  
; CHECK-NEXT:  ret  
    %c = mul i32 %a, 320  
    ret i32 %c  
}
```

```
define signext i32 @mulw576(i32 signext %a) {  
; CHECK-LABEL: mulw576:  
; CHECK:    # %bb.0:  
; CHECK-NEXT:  li a1, 576  
; CHECK-NEXT:  mulw a0, a0, a1  
; CHECK-NEXT:  ret  
    %c = mul i32 %a, 576  
    ret i32 %c  
}
```

```
}
```

```
define i64 @add4104(i64 %a) {  
; RV64I-LABEL: add4104:  
; RV64I:  
    # %bb.0:  
; RV64I-NEXT:    lui a1, 1  
; RV64I-NEXT:    addiw a1, a1, 8  
; RV64I-NEXT:    add a0, a0, a1  
; RV64I-NEXT:    ret  
;  
; RV64ZBA-LABEL: add4104:  
; RV64ZBA:    # %bb.0:  
; RV64ZBA-NEXT:    li a1, 1026  
; RV64ZBA-NEXT:    sh2add a0, a1, a0  
; RV64ZBA-NEXT:    ret  
    %c = add i64 %a, 4104  
    ret i64 %c  
}
```

```
define i64 @add8208(i64 %a) {  
; RV64I-LABEL: add8208:  
; RV64I:    # %bb.0:  
; RV64I-NEXT:    lui a1, 2  
; RV64I-NEXT:    addiw a1, a1, 16  
; RV64I-NEXT:    add a0, a0, a1  
; RV64I-NEXT:    ret  
;  
; RV64ZBA-LABEL: add8208:  
; RV64ZBA:    # %bb.0:  
; RV64ZBA-NEXT:    li a1, 1026  
; RV64ZBA-NEXT:    sh3add a0, a1, a0  
; RV64ZBA-NEXT:    ret  
    %c = add i64 %a, 8208  
    ret i64 %c  
}
```

; Make sure we prefer LUI for the 8192 instead of using sh3add.

```
define signext i32 @add8192_i32(i32 signext %a) {  
; CHECK-LABEL: add8192_i32:  
; CHECK:    # %bb.0:  
; CHECK-NEXT:    lui a1, 2  
; CHECK-NEXT:    addw a0, a0, a1  
; CHECK-NEXT:    ret  
    %c = add i32 %a, 8192  
    ret i32 %c  
}
```

```

; Make sure we prefer LUI
for the 8192 instead of using sh3add.
define i64 @add8192(i64 %a) {
; CHECK-LABEL: add8192:
; CHECK:      # %bb.0:
; CHECK-NEXT:  lui a1, 2
; CHECK-NEXT:  add a0, a0, a1
; CHECK-NEXT:  ret
%c = add i64 %a, 8192
ret i64 %c
}

define signext i32 @addshl32_5_6(i32 signext %a, i32 signext %b) {
; RV64I-LABEL: addshl32_5_6:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 5
; RV64I-NEXT:  slli a1, a1, 6
; RV64I-NEXT:  addw a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addshl32_5_6:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT: sh1add a0, a1, a0
; RV64ZBA-NEXT: slliw a0, a0, 5
; RV64ZBA-NEXT: ret
%c = shl i32 %a, 5
%d = shl i32 %b, 6
%e = add i32 %c, %d
ret i32 %e
}

define i64 @addshl64_5_6(i64 %a, i64 %b) {
; RV64I-LABEL: addshl64_5_6:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 5
; RV64I-NEXT:  slli a1, a1, 6
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addshl64_5_6:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:
    sh1add a0, a1, a0
; RV64ZBA-NEXT: slli a0, a0, 5
; RV64ZBA-NEXT: ret
%c = shl i64 %a, 5
%d = shl i64 %b, 6
%e = add i64 %c, %d

```

```

ret i64 %e
}

define signext i32 @addshl32_5_7(i32 signext %a, i32 signext %b) {
; RV64I-LABEL: addshl32_5_7:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 5
; RV64I-NEXT:  slli a1, a1, 7
; RV64I-NEXT:  addw a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addshl32_5_7:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh2add a0, a1, a0
; RV64ZBA-NEXT:  slliw a0, a0, 5
; RV64ZBA-NEXT:  ret
%c = shl i32 %a, 5
%d = shl i32 %b, 7
%e = add i32 %c, %d
ret i32 %e
}

```

```

define i64 @addshl64_5_7(i64 %a, i64 %b) {
; RV64I-LABEL: addshl64_5_7:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 5
; RV64I-NEXT:  slli a1, a1, 7
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addshl64_5_7:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh2add a0, a1, a0
; RV64ZBA-NEXT:  slli a0, a0, 5
; RV64ZBA-NEXT:  ret
%c
= shl i64 %a, 5
%d = shl i64 %b, 7
%e = add i64 %c, %d
ret i64 %e
}

```

```

define signext i32 @addshl32_5_8(i32 signext %a, i32 signext %b) {
; RV64I-LABEL: addshl32_5_8:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 5
; RV64I-NEXT:  slli a1, a1, 8
; RV64I-NEXT:  addw a0, a0, a1

```

```

; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addshl32_5_8:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh3add a0, a1, a0
; RV64ZBA-NEXT:  slliw a0, a0, 5
; RV64ZBA-NEXT:  ret
%c = shl i32 %a, 5
%d = shl i32 %b, 8
%e = add i32 %c, %d
ret i32 %e
}

```

```

define i64 @addshl64_5_8(i64 %a, i64 %b) {
; RV64I-LABEL: addshl64_5_8:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 5
; RV64I-NEXT:  slli a1, a1, 8
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: addshl64_5_8:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  sh3add a0, a1, a0
; RV64ZBA-NEXT:  slli a0, a0, 5
; RV64ZBA-NEXT:  ret
%c = shl i64 %a, 5
%d = shl i64 %b, 8
%e = add i64 %c, %d
ret i64 %e
}

```

```

; Make
sure we use sext.h+slli+srli for Zba+Zbb.
; FIXME: The RV64I and Zba only cases can be done with only 3 shifts.

```

```

define zeroext i32 @sext_ashr_zext_i8(i8 %a) nounwind {

```

```

; RV64I-LABEL: sext_ashr_zext_i8:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 24
; RV64I-NEXT:  sraiw a0, a0, 31
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  ret
;
; RV64ZBANOZBB-LABEL: sext_ashr_zext_i8:
; RV64ZBANOZBB:      # %bb.0:
; RV64ZBANOZBB-NEXT:  slli a0, a0, 24
; RV64ZBANOZBB-NEXT:  sraiw a0, a0, 31

```

```

; RV64ZBANOBZBB-NEXT:  zext.w a0, a0
; RV64ZBANOBZBB-NEXT:  ret
;
; RV64ZBAZBB-LABEL: sext_ashr_zext_i8:
; RV64ZBAZBB:          # %bb.0:
; RV64ZBAZBB-NEXT:     sext.b a0, a0
; RV64ZBAZBB-NEXT:     srai.w a0, a0, 9
; RV64ZBAZBB-NEXT:     zext.w a0, a0
; RV64ZBAZBB-NEXT:     ret
%ext = sext i8 %a to i32
%1 = ashr i32 %ext, 9
ret i32 %1
}

```

; Make sure we use sext.h+slli+srli for Zba+Zbb.  
; FIXME: The RV64I and Zba only cases can be done with only 3 shifts.  
define

```

zeroext i32 @sext_ashr_zext_i16(i16 %a) nounwind {
; RV64I-LABEL: sext_ashr_zext_i16:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 16
; RV64I-NEXT:  srai.w a0, a0, 25
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  ret
;
; RV64ZBANOBZBB-LABEL: sext_ashr_zext_i16:
; RV64ZBANOBZBB:      # %bb.0:
; RV64ZBANOBZBB-NEXT:  slli a0, a0, 16
; RV64ZBANOBZBB-NEXT:  srai.w a0, a0, 25
; RV64ZBANOBZBB-NEXT:  zext.w a0, a0
; RV64ZBANOBZBB-NEXT:  ret
;
; RV64ZBAZBB-LABEL: sext_ashr_zext_i16:
; RV64ZBAZBB:      # %bb.0:
; RV64ZBAZBB-NEXT:  slli a0, a0, 48
; RV64ZBAZBB-NEXT:  srai a0, a0, 57
; RV64ZBAZBB-NEXT:  zext.w a0, a0
; RV64ZBAZBB-NEXT:  ret
%ext = sext i16 %a to i32
%1 = ashr i32 %ext, 9
ret i32 %1
}

```

; This the IR you get from InstCombine if take the difference of 2 pointers and  
; cast is to unsigned before using as an index.

```

define signext i16 @sh1adduw_ptrdiff(i64 %diff, ptr %baseptr) {
; RV64I-LABEL: sh1adduw_ptrdiff:

```



```

; RV64I:
    # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  slli a2, a2, 33
; RV64I-NEXT:  addi a2, a2, -2
; RV64I-NEXT:  and a0, a0, a2
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  lh a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh1adduw_ptrdiff:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  srli a0, a0, 1
; RV64ZBA-NEXT:  sh1add.uw a0, a0, a1
; RV64ZBA-NEXT:  lh a0, 0(a0)
; RV64ZBA-NEXT:  ret
%ptrdiff = lshr exact i64 %diff, 1
%cast = and i64 %ptrdiff, 4294967295
%ptr = getelementptr inbounds i16, ptr %baseptr, i64 %cast
%res = load i16, ptr %ptr
ret i16 %res
}

define signext i32 @sh2adduw_ptrdiff(i64 %diff, ptr %baseptr) {
; RV64I-LABEL: sh2adduw_ptrdiff:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  slli a2, a2, 34
; RV64I-NEXT:  addi a2, a2, -4
; RV64I-NEXT:  and a0, a0, a2
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  lw a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh2adduw_ptrdiff:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:
    srli a0, a0, 2
; RV64ZBA-NEXT:  sh2add.uw a0, a0, a1
; RV64ZBA-NEXT:  lw a0, 0(a0)
; RV64ZBA-NEXT:  ret
%ptrdiff = lshr exact i64 %diff, 2
%cast = and i64 %ptrdiff, 4294967295
%ptr = getelementptr inbounds i32, ptr %baseptr, i64 %cast
%res = load i32, ptr %ptr
ret i32 %res
}

```

```

define i64 @sh3adduw_ptrdiff(i64 %diff, ptr %baseptr) {
; RV64I-LABEL: sh3adduw_ptrdiff:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a2, 1
; RV64I-NEXT:  slli a2, a2, 35
; RV64I-NEXT:  addi a2, a2, -8
; RV64I-NEXT:  and a0, a0, a2
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  ld a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: sh3adduw_ptrdiff:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  srli a0, a0, 3
; RV64ZBA-NEXT:  sh3add.uw a0, a0, a1
; RV64ZBA-NEXT:  ld a0, 0(a0)
; RV64ZBA-NEXT:  ret
%ptrdiff = lshr exact i64 %diff, 3
%cast = and i64 %ptrdiff, 4294967295
%ptr = getelementptr inbounds i64, ptr %baseptr, i64 %cast
%res = load i64, ptr %ptr
ret i64 %res
}

```

```

define signext
i16 @srlw_1_sh1add(ptr %0, i32 signext %1) {
; RV64I-LABEL: srlw_1_sh1add:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srlw a1, a1, 1
; RV64I-NEXT:  slli a1, a1, 1
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  lh a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: srlw_1_sh1add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  srlw a1, a1, 1
; RV64ZBA-NEXT:  sh1add a0, a1, a0
; RV64ZBA-NEXT:  lh a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = lshr i32 %1, 1
%4 = zext i32 %3 to i64
%5 = getelementptr inbounds i16, ptr %0, i64 %4
%6 = load i16, ptr %5, align 2
ret i16 %6
}

```

```

define i128 @slliuw_ptrdiff(i64 %diff, ptr %baseptr) {

```

```

; RV64I-LABEL: slliuw_ptrdiff:
; RV64I:      # %bb.0:
; RV64I-NEXT: li a2, 1
; RV64I-NEXT: slli a2, a2, 36
; RV64I-NEXT: addi a2, a2, -16
; RV64I-NEXT: and a0, a0, a2
; RV64I-NEXT: add a1, a1, a0
; RV64I-NEXT: ld a0, 0(a1)
; RV64I-NEXT: ld a1, 8(a1)
; RV64I-NEXT: ret
;
; RV64ZBA-LABEL: slliuw_ptrdiff:
; RV64ZBA:      # %bb.0:
;
RV64ZBA-NEXT: srli a0, a0, 4
; RV64ZBA-NEXT: slli.uw a0, a0, 4
; RV64ZBA-NEXT: add a1, a1, a0
; RV64ZBA-NEXT: ld a0, 0(a1)
; RV64ZBA-NEXT: ld a1, 8(a1)
; RV64ZBA-NEXT: ret
%ptrdiff = lshr exact i64 %diff, 4
%cast = and i64 %ptrdiff, 4294967295
%ptr = getelementptr inbounds i128, ptr %baseptr, i64 %cast
%res = load i128, ptr %ptr
ret i128 %res
}

```

```

define signext i32 @srliw_2_sh2add(ptr %0, i32 signext %1) {
; RV64I-LABEL: srliw_2_sh2add:
; RV64I:      # %bb.0:
; RV64I-NEXT: srliw a1, a1, 2
; RV64I-NEXT: slli a1, a1, 2
; RV64I-NEXT: add a0, a0, a1
; RV64I-NEXT: lw a0, 0(a0)
; RV64I-NEXT: ret
;
; RV64ZBA-LABEL: srliw_2_sh2add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT: srliw a1, a1, 2
; RV64ZBA-NEXT: sh2add a0, a1, a0
; RV64ZBA-NEXT: lw a0, 0(a0)
; RV64ZBA-NEXT: ret
%3 = lshr i32 %1, 2
%4 = zext i32 %3 to i64
%5 = getelementptr inbounds i32, ptr %0, i64 %4
%6 = load i32, ptr %5, align 4
ret i32 %6
}

```

```

define i64 @srliw_3_sh3add(ptr
    %0, i32 signext %1) {
; RV64I-LABEL: srliw_3_sh3add:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srliw a1, a1, 3
; RV64I-NEXT:  slli a1, a1, 3
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ld a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: srliw_3_sh3add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  srliw a1, a1, 3
; RV64ZBA-NEXT:  sh3add a0, a1, a0
; RV64ZBA-NEXT:  ld a0, 0(a0)
; RV64ZBA-NEXT:  ret
    %3 = lshr i32 %1, 3
    %4 = zext i32 %3 to i64
    %5 = getelementptr inbounds i64, ptr %0, i64 %4
    %6 = load i64, ptr %5, align 8
    ret i64 %6
}

define signext i32 @srliw_1_sh2add(ptr %0, i32 signext %1) {
; RV64I-LABEL: srliw_1_sh2add:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srliw a1, a1, 1
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  srli a1, a1, 30
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  lw a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: srliw_1_sh2add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  srliw a1, a1, 1
; RV64ZBA-NEXT:  sh2add a0, a1, a0
;
    RV64ZBA-NEXT:  lw a0, 0(a0)
; RV64ZBA-NEXT:  ret
    %3 = lshr i32 %1, 1
    %4 = zext i32 %3 to i64
    %5 = getelementptr inbounds i32, ptr %0, i64 %4
    %6 = load i32, ptr %5, align 4
    ret i32 %6
}

```

```

define i64 @srliw_1_sh3add(ptr %0, i32 signext %1) {
; RV64I-LABEL: srliw_1_sh3add:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srliw a1, a1, 1
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  srli a1, a1, 29
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ld a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: srliw_1_sh3add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  srliw a1, a1, 1
; RV64ZBA-NEXT:  sh3add a0, a1, a0
; RV64ZBA-NEXT:  ld a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = lshr i32 %1, 1
%4 = zext i32 %3 to i64
%5 = getelementptr inbounds i64, ptr %0, i64 %4
%6 = load i64, ptr %5, align 8
ret i64 %6
}

```

```

define i64 @srliw_2_sh3add(ptr %0, i32 signext %1) {
; RV64I-LABEL: srliw_2_sh3add:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srliw a1, a1, 2
; RV64I-NEXT:  slli a1, a1, 32
;
; RV64I-NEXT:  srli a1, a1, 29
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ld a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: srliw_2_sh3add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  srliw a1, a1, 2
; RV64ZBA-NEXT:  sh3add a0, a1, a0
; RV64ZBA-NEXT:  ld a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = lshr i32 %1, 2
%4 = zext i32 %3 to i64
%5 = getelementptr inbounds i64, ptr %0, i64 %4
%6 = load i64, ptr %5, align 8
ret i64 %6
}

```

```

define signext i16 @srliw_2_sh1add(ptr %0, i32 signext %1) {

```

```

; RV64I-LABEL: srlw_2_sh1add:
; RV64I:    # %bb.0:
; RV64I-NEXT:  srlw a1, a1, 2
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  srli a1, a1, 31
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  lh a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: srlw_2_sh1add:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  srlw a1, a1, 2
; RV64ZBA-NEXT:  sh1add a0, a1, a0
; RV64ZBA-NEXT:  lh a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = lshr i32 %1, 2
%4 = zext i32 %3 to i64
%5 = getelementptr
inbounds i16, ptr %0, i64 %4
%6 = load i16, ptr %5, align 2
ret i16 %6
}

define signext i32 @srlw_3_sh2add(ptr %0, i32 signext %1) {
; RV64I-LABEL: srlw_3_sh2add:
; RV64I:    # %bb.0:
; RV64I-NEXT:  srlw a1, a1, 3
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  srli a1, a1, 30
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  lw a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: srlw_3_sh2add:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  srlw a1, a1, 3
; RV64ZBA-NEXT:  sh2add a0, a1, a0
; RV64ZBA-NEXT:  lw a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = lshr i32 %1, 3
%4 = zext i32 %3 to i64
%5 = getelementptr inbounds i32, ptr %0, i64 %4
%6 = load i32, ptr %5, align 4
ret i32 %6
}

define i64 @srlw_4_sh3add(ptr %0, i32 signext %1) {

```

```

; RV64I-LABEL: srlw_4_sh3add:
; RV64I:      # %bb.0:
; RV64I-NEXT: srlw a1, a1, 4
; RV64I-NEXT: slli a1, a1, 32
; RV64I-NEXT: srli a1, a1, 29
; RV64I-NEXT: add a0, a0, a1
; RV64I-NEXT: ld a0, 0(a0)
; RV64I-NEXT:
    ret
;
; RV64ZBA-LABEL: srlw_4_sh3add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT: srlw a1, a1, 4
; RV64ZBA-NEXT: sh3add a0, a1, a0
; RV64ZBA-NEXT: ld a0, 0(a0)
; RV64ZBA-NEXT: ret
%3 = lshr i32 %1, 4
%4 = zext i32 %3 to i64
%5 = getelementptr inbounds i64, ptr %0, i64 %4
%6 = load i64, ptr %5, align 8
ret i64 %6
}

```

```

define signext i32 @srli_1_sh2add(ptr %0, i64 %1) {
; RV64I-LABEL: srli_1_sh2add:
; RV64I:      # %bb.0:
; RV64I-NEXT: slli a1, a1, 1
; RV64I-NEXT: andi a1, a1, -4
; RV64I-NEXT: add a0, a0, a1
; RV64I-NEXT: lw a0, 0(a0)
; RV64I-NEXT: ret
;
; RV64ZBA-LABEL: srli_1_sh2add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT: srli a1, a1, 1
; RV64ZBA-NEXT: sh2add a0, a1, a0
; RV64ZBA-NEXT: lw a0, 0(a0)
; RV64ZBA-NEXT: ret
%3 = lshr i64 %1, 1
%4 = getelementptr inbounds i32, ptr %0, i64 %3
%5 = load i32, ptr %4, align 4
ret i32 %5
}

```

```

define i64 @srli_2_sh3add(ptr %0, i64 %1) {
; RV64I-LABEL: srli_2_sh3add:
; RV64I:      # %bb.0:
;

```

```

RV64I-NEXT:  slli a1, a1, 1
; RV64I-NEXT:  andi a1, a1, -8
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ld a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: srli_2_sh3add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  srli a1, a1, 2
; RV64ZBA-NEXT:  sh3add a0, a1, a0
; RV64ZBA-NEXT:  ld a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = lshr i64 %1, 2
%4 = getelementptr inbounds i64, ptr %0, i64 %3
%5 = load i64, ptr %4, align 8
ret i64 %5
}

define signext i16 @srli_2_sh1add(ptr %0, i64 %1) {
; RV64I-LABEL: srli_2_sh1add:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srli a1, a1, 1
; RV64I-NEXT:  andi a1, a1, -2
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  lh a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: srli_2_sh1add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  srli a1, a1, 2
; RV64ZBA-NEXT:  sh1add a0, a1, a0
; RV64ZBA-NEXT:  lh a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = lshr i64 %1, 2
%4 = getelementptr inbounds i16, ptr %0, i64 %3
%5 = load i16, ptr %4, align
2
ret i16 %5
}

define signext i32 @srli_3_sh2add(ptr %0, i64 %1) {
; RV64I-LABEL: srli_3_sh2add:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srli a1, a1, 1
; RV64I-NEXT:  andi a1, a1, -4
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  lw a0, 0(a0)
; RV64I-NEXT:  ret

```



```

;
; RV64ZBA-LABEL: srli_3_sh2add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT: srli a1, a1, 3
; RV64ZBA-NEXT: sh2add a0, a1, a0
; RV64ZBA-NEXT: lw a0, 0(a0)
; RV64ZBA-NEXT: ret
%3 = lshr i64 %1, 3
%4 = getelementptr inbounds i32, ptr %0, i64 %3
%5 = load i32, ptr %4, align 4
ret i32 %5
}

```

```

define i64 @srli_4_sh3add(ptr %0, i64 %1) {
; RV64I-LABEL: srli_4_sh3add:
; RV64I:      # %bb.0:
; RV64I-NEXT: srli a1, a1, 1
; RV64I-NEXT: andi a1, a1, -8
; RV64I-NEXT: add a0, a0, a1
; RV64I-NEXT: ld a0, 0(a0)
; RV64I-NEXT: ret
;
; RV64ZBA-LABEL: srli_4_sh3add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT: srli a1, a1, 4
; RV64ZBA-NEXT: sh3add a0, a1, a0
; RV64ZBA-NEXT: ld a0, 0(a0)
; RV64ZBA-NEXT:
ret
%3 = lshr i64 %1, 4
%4 = getelementptr inbounds i64, ptr %0, i64 %3
%5 = load i64, ptr %4, align 8
ret i64 %5
}

```

```

define signext i16 @shl_2_sh1add(ptr %0, i32 signext %1) {
; RV64I-LABEL: shl_2_sh1add:
; RV64I:      # %bb.0:
; RV64I-NEXT: slli a1, a1, 2
; RV64I-NEXT: slli a1, a1, 32
; RV64I-NEXT: srli a1, a1, 31
; RV64I-NEXT: add a0, a0, a1
; RV64I-NEXT: lh a0, 0(a0)
; RV64I-NEXT: ret
;
; RV64ZBA-LABEL: shl_2_sh1add:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT: slli a1, a1, 2

```

```

; RV64ZBA-NEXT:  zext.w a1, a1
; RV64ZBA-NEXT:  sh1add a0, a1, a0
; RV64ZBA-NEXT:  lh a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = shl i32 %1, 2
%4 = zext i32 %3 to i64
%5 = getelementptr inbounds i16, ptr %0, i64 %4
%6 = load i16, ptr %5, align 2
ret i16 %6
}

```

```

define signext i32 @shl_16_sh2add(ptr %0, i32 signext %1) {
; RV64I-LABEL: shl_16_sh2add:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a1, a1, 16
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  srli a1, a1, 30
;
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  lw a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: shl_16_sh2add:
; RV64ZBA:    # %bb.0:
; RV64ZBA-NEXT:  slli a1, a1, 16
; RV64ZBA-NEXT:  zext.w a1, a1
; RV64ZBA-NEXT:  sh2add a0, a1, a0
; RV64ZBA-NEXT:  lw a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = shl i32 %1, 16
%4 = zext i32 %3 to i64
%5 = getelementptr inbounds i32, ptr %0, i64 %4
%6 = load i32, ptr %5, align 4
ret i32 %6
}

```

```

define i64 @shl_31_sh3add(ptr %0, i32 signext %1) {
; RV64I-LABEL: shl_31_sh3add:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a1, a1, 31
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  srli a1, a1, 29
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ld a0, 0(a0)
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: shl_31_sh3add:
; RV64ZBA:    # %bb.0:

```

```

; RV64ZBA-NEXT:  slli a1, a1, 31
; RV64ZBA-NEXT:  zext.w a1, a1
; RV64ZBA-NEXT:  sh3add a0, a1, a0
; RV64ZBA-NEXT:  ld a0, 0(a0)
; RV64ZBA-NEXT:  ret
%3 = shl i32 %1, 31
%4 = zext i32 %3
to i64
%5 = getelementptr inbounds i64, ptr %0, i64 %4
%6 = load i64, ptr %5, align 8
ret i64 %6
}

```

```

define i64 @pack_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: pack_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: pack_i64:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  slli a1, a1, 32
; RV64ZBA-NEXT:  add.uw a0, a0, a1
; RV64ZBA-NEXT:  ret
%shl = and i64 %a, 4294967295
%shl1 = shl i64 %b, 32
%or = or i64 %shl1, %shl
ret i64 %or
}

```

```

define i64 @pack_i64_2(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: pack_i64_2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: pack_i64_2:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  slli a1, a1, 32
; RV64ZBA-NEXT:  add.uw a0, a0,
a1
; RV64ZBA-NEXT:  ret

```

```

%zexta = zext i32 %a to i64
%zextb = zext i32 %b to i64
%shl1 = shl i64 %zextb, 32
%or = or i64 %shl1, %zexta
ret i64 %or
}

```

```

define i64 @pack_i64_3(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: pack_i64_3:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi a0, a0, 1
; RV64I-NEXT:  addi a1, a1, 1
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBA-LABEL: pack_i64_3:
; RV64ZBA:      # %bb.0:
; RV64ZBA-NEXT:  addi a0, a0, 1
; RV64ZBA-NEXT:  addi a1, a1, 1
; RV64ZBA-NEXT:  slli a1, a1, 32
; RV64ZBA-NEXT:  add.uw a0, a0, a1
; RV64ZBA-NEXT:  ret
%adda = add i32 %a, 1
%addb = add i32 %b, 1
%zexta = zext i32 %adda to i64
%zextb = zext i32 %addb to i64
%shl1 = shl i64 %zextb, 32
%or = or i64 %shl1, %zexta
ret i64 %or
}

```

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
 Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative

Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of

the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at



<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

=====  
Software from third parties included in the LLVM Project:  
=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or `LICENSE` file at the top containing the specific license and restrictions which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every file.

=====  
Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):  
=====

The software contained in this directory tree is dual licensed under both the University of Illinois "BSD-Like" license and the MIT license. As a user of this code you may choose to use it under either license. As a contributor, you agree to allow your code to be used under both.

Full text of the relevant licenses is included below.

=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2017-2019 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:  
Threading Runtimes Team  
Intel Corporation  
<http://www.intel.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- \* Neither the names of Intel Corporation Threading Runtimes Team nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====

Copyright (c) 2017-2019 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

IMath is Copyright 2002-2009 Michael J. Fromberger  
You may use it subject to the following Licensing Terms:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py  
; RUN: llc -mtriple=riscv64 -mattr=+zbb -verify-machineinstrs < %s \  
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64ZBB
```

```
declare i32 @llvm.riscv.orc.b.i32(i32)
```

```
define signext i32 @orc32(i32 signext %a) nounwind {  
; RV64ZBB-LABEL: orc32:  
; RV64ZBB:      # %bb.0:  
; RV64ZBB-NEXT:  orc.b a0, a0
```

```
; RV64ZBB-NEXT:  sext.w a0, a0
; RV64ZBB-NEXT:  ret
%tmp = call i32 @llvm.riscv.orc.b.i32(i32 %a)
ret i32 %tmp
}
```

```
define zeroext i32 @orc32_zext(i32 zeroext %a) nounwind {
; RV64ZBB-LABEL: orc32_zext:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  orc.b a0, a0
; RV64ZBB-NEXT:  ret
%tmp = call i32 @llvm.riscv.orc.b.i32(i32 %a)
ret i32 %tmp
}
```

; Second and+or is redundant with the first, make sure we remove them.

```
define signext i32 @orc32_knownbits(i32 signext %a) nounwind {
; RV64ZBB-LABEL: orc32_knownbits:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  lui a1, 1044480
; RV64ZBB-NEXT:
    and a0, a0, a1
; RV64ZBB-NEXT:  lui a1, 2048
; RV64ZBB-NEXT:  addi a1, a1, 1
; RV64ZBB-NEXT:  or a0, a0, a1
; RV64ZBB-NEXT:  orc.b a0, a0
; RV64ZBB-NEXT:  sext.w a0, a0
; RV64ZBB-NEXT:  ret
%tmp = and i32 %a, 4278190080 ; 0xFF000000
%tmp2 = or i32 %tmp, 8388609 ; 0x800001
%tmp3 = call i32 @llvm.riscv.orc.b.i32(i32 %tmp2)
%tmp4 = and i32 %tmp3, 4278190080 ; 0xFF000000
%tmp5 = or i32 %tmp4, 16711935 ; 0xFF00FF
ret i32 %tmp5
}
```

```
declare i64 @llvm.riscv.orc.b.i64(i64)
```

```
define i64 @orc64(i64 %a) nounwind {
; RV64ZBB-LABEL: orc64:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  orc.b a0, a0
; RV64ZBB-NEXT:  ret
%tmp = call i64 @llvm.riscv.orc.b.i64(i64 %a)
ret i64 %tmp
}
```

; Second and+or is redundant with the first, make sure we remove them.

```

define i64 @orcb64_knownbits(i64 %a) nounwind {
; RV64ZBB-LABEL: orcb64_knownbits:
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT: lui a1, 65535
; RV64ZBB-NEXT: slli a1, a1, 12
; RV64ZBB-NEXT: and a0, a0, a1
;
; RV64ZBB-NEXT: lui a1, 256
; RV64ZBB-NEXT: addiw a1, a1, 8
; RV64ZBB-NEXT: slli a2, a1, 42
; RV64ZBB-NEXT: add a1, a1, a2
; RV64ZBB-NEXT: or a0, a0, a1
; RV64ZBB-NEXT: orc.b a0, a0
; RV64ZBB-NEXT: ret
%tmp = and i64 %a, 1099494850560 ; 0x000000ffff000000
%tmp2 = or i64 %tmp, 4611721202800525320 ; 0x4000200000100008
%tmp3 = call i64 @llvm.riscv.orc.b.i64(i64 %tmp2)
%tmp4 = and i64 %tmp3, 1099494850560 ; 0x000000ffff000000
%tmp5 = or i64 %tmp4, 18374966855153418495 ; 0xff00ff0000ff00ff
ret i64 %tmp5
}
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -mattr=+zbc -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64ZBC

```

```

declare i64 @llvm.riscv.clmulr.i64(i64 %a, i64 %b)

```

```

define i64 @clmul64r(i64 %a, i64 %b) nounwind {
; RV64ZBC-LABEL: clmul64r:
; RV64ZBC:      # %bb.0:
; RV64ZBC-NEXT: clmulr a0, a0, a1
; RV64ZBC-NEXT: ret
%tmp = call i64 @llvm.riscv.clmulr.i64(i64 %a, i64 %b)
ret i64 %tmp
}

```

```

declare i32 @llvm.riscv.clmulr.i32(i32 %a, i32 %b)

```

```

define signext i32 @clmul32r(i32 signext %a, i32 signext %b) nounwind {
; RV64ZBC-LABEL: clmul32r:
; RV64ZBC:      # %bb.0:
; RV64ZBC-NEXT: slli a1, a1, 32
; RV64ZBC-NEXT: slli a0, a0, 32
; RV64ZBC-NEXT: clmulr a0, a0, a1
; RV64ZBC-NEXT: srai a0, a0, 32
; RV64ZBC-NEXT: ret
%tmp = call i32 @llvm.riscv.clmulr.i32(i32 %a, i32 %b)
ret i32 %tmp

```

```

}

; FIXME: We could avoid the slli instructions by using clmul+srli+sext.w
since
; the inputs are zero extended.
define signext i32 @clmul32r_zext(i32 zeroext %a, i32 zeroext %b) nounwind {
; RV64ZBC-LABEL: clmul32r_zext:
; RV64ZBC:      # %bb.0:
; RV64ZBC-NEXT: slli a1, a1, 32
; RV64ZBC-NEXT: slli a0, a0, 32
; RV64ZBC-NEXT: clmulr a0, a0, a1
; RV64ZBC-NEXT: srli a0, a0, 32
; RV64ZBC-NEXT: ret
    %tmp = call i32 @llvm.riscv.clmulr.i32(i32 %a, i32 %b)
    ret i32 %tmp
}

; RUN: opt %s -passes=inline -S | FileCheck %s

define internal void @innerSmall() "min-legal-vector-width"="128" {
    ret void
}

define internal void @innerLarge() "min-legal-vector-width"="512" {
    ret void
}

define internal void @innerNoAttribute() {
    ret void
}

; We should not add an attribute during inlining. No attribute means unknown.
; Inlining doesn't change the fact that we don't know anything about this
; function.
define void @outerNoAttribute() {
    call void @innerLarge()
    ret void
}

define void @outerConflictingAttributeSmall() "min-legal-vector-width"="128" {
    call void @innerLarge()
    ret void
}

define void @outerConflictingAttributeLarge() "min-legal-vector-width"="512" {
    call void @innerSmall()
    ret void
}

```

```

; We should remove the attribute after inlining since the callee's
; vector width requirements are unknown.
define void @outerAttribute() "min-legal-vector-width"="128" {
  call void @innerNoAttribute()
  ret void
}

; CHECK: define void @outerNoAttribute()
{
; CHECK: define void @outerConflictingAttributeSmall() #0
; CHECK: define void @outerConflictingAttributeLarge() #0
; CHECK: define void @outerAttribute() {
; CHECK: attributes #0 = { "min-legal-vector-width"="512" }
; RUN: opt -mtriple=aarch64-linux-gnu -mattr=+sve -passes=sanitize-masked-mem-intrin -S < %s | FileCheck %s

; Testing that masked scatters operating on scalable vectors that are
; packed in SVE registers are not scalarized.

; CHECK-LABEL: @masked_scatter_nxv4i32(
; CHECK: call void @llvm.masked.scatter.nxv4i32
define void @masked_scatter_nxv4i32(<vscale x 4 x i32> %data, <vscale x 4 x ptr> %ptrs, <vscale x 4 x i1>
%mask) {
  call void @llvm.masked.scatter.nxv4i32(<vscale x 4 x i32> %data, <vscale x 4 x ptr> %ptrs, i32 0, <vscale x 4 x
i1> %mask)
  ret void
}

; Testing that masked scatters operating on scalable vectors of FP
; data that is packed in SVE registers are not scalarized.

; CHECK-LABEL: @masked_scatter_nxv2f64(
; CHECK: call void @llvm.masked.scatter.nxv2f64
define void @masked_scatter_nxv2f64(<vscale x 2 x double> %data, <vscale x 2 x ptr> %ptrs, <vscale x 2 x i1>
%mask) {
  call void @llvm.masked.scatter.nxv2f64(<vscale x 2 x double> %data, <vscale x 2 x ptr> %ptrs,
i32 0, <vscale x 2 x i1> %mask)
  ret void
}

; Testing that masked scatters operating on scalable vectors of FP
; data that is unpacked in SVE registers are not scalarized.

; CHECK-LABEL: @masked_scatter_nxv2f16(
; CHECK: call void @llvm.masked.scatter.nxv2f16
define void @masked_scatter_nxv2f16(<vscale x 2 x half> %data, <vscale x 2 x ptr> %ptrs, <vscale x 2 x i1>
%mask) {
  call void @llvm.masked.scatter.nxv2f16(<vscale x 2 x half> %data, <vscale x 2 x ptr> %ptrs, i32 0, <vscale x 2 x
i1> %mask)

```

```

ret void
}

; Testing that masked scatters operating on 64-bit fixed vectors are
; scalarized because NEON doesn't have support for masked scatter
; instructions.

; CHECK-LABEL: @masked_scatter_v2f32(
; CHECK-NOT: @llvm.masked.scatter.v2f32(
define void @masked_scatter_v2f32(<2 x float> %data, <2 x ptr> %ptrs, <2 x i1> %masks) {
    call void @llvm.masked.scatter.v2f32(<2 x float> %data, <2 x ptr> %ptrs, i32 0, <2 x i1> %masks)
    ret void
}

; Testing that masked scatters operating on 128-bit
; fixed vectors are
; scalarized because NEON doesn't have support for masked scatter
; instructions and because we are not targeting fixed width SVE.

; CHECK-LABEL: @masked_scatter_v4i32(
; CHECK-NOT: @llvm.masked.scatter.v4i32(
define void @masked_scatter_v4i32(<4 x i32> %data, <4 x ptr> %ptrs, <4 x i1> %masks) {
    call void @llvm.masked.scatter.v4i32(<4 x i32> %data, <4 x ptr> %ptrs, i32 0, <4 x i1> %masks)
    ret void
}

declare void @llvm.masked.scatter.nxv4i32(<vscale x 4 x i32> %data, <vscale x 4 x ptr> %ptrs, i32 %align,
<vscale x 4 x i1> %masks)
declare void @llvm.masked.scatter.nxv2f64(<vscale x 2 x double> %data, <vscale x 2 x ptr> %ptrs, i32 %align,
<vscale x 2 x i1> %masks)
declare void @llvm.masked.scatter.nxv2f16(<vscale x 2 x half> %data, <vscale x 2 x ptr> %ptrs, i32 %align,
<vscale x 2 x i1> %masks)
declare void @llvm.masked.scatter.v2f32(<2 x float> %data, <2 x ptr> %ptrs, i32 %align, <2 x i1> %masks)
declare void @llvm.masked.scatter.v4i32(<4 x i32> %data, <4 x
ptr> %ptrs, i32 %align, <4 x i1> %masks)
//===-- ObjectContainerBSDArchive.h -----*- C++ -*-====//
//
// Part of the LLVM Project, under the Apache License v2.0 with LLVM Exceptions.
// See https://llvm.org/LICENSE.txt for license information.
// SPDX-License-Identifier: Apache-2.0 WITH LLVM-exception
//
//===-----=====//

#ifndef
LLDB_SOURCE_PLUGINS_OBJECTCONTAINER_BSD_ARCHIVE_OBJECTCONTAINERBSDARCHIVE_H
#define
LLDB_SOURCE_PLUGINS_OBJECTCONTAINER_BSD_ARCHIVE_OBJECTCONTAINERBSDARCHIVE_H

```



```

#include "lldb/Core/UniqueCStringMap.h"
#include "lldb/Symbol/ObjectContainer.h"
#include "lldb/Utility/ArchSpec.h"
#include "lldb/Utility/ConstString.h"
#include "lldb/Utility/FileSpec.h"

#include "llvm/Object/Archive.h"
#include "llvm/Support/Chrono.h"
#include "llvm/Support/Path.h"

#include <map>
#include <memory>
#include <mutex>

enum class ArchiveType { Invalid, Archive, ThinArchive };

class ObjectContainerBSDArchive : public lldb_private::ObjectContainer
{
public:
    ObjectContainerBSDArchive(const lldb::ModuleSP &module_sp,
                             lldb::DataBufferSP &data_sp,
                             lldb::offset_t data_offset,
                             const lldb_private::FileSpec *file,
                             lldb::offset_t offset, lldb::offset_t length,
                             ArchiveType archive_type);

    ~ObjectContainerBSDArchive() override;

    // Static Functions
    static void Initialize();

    static void Terminate();

    static llvm::StringRef GetPluginNameStatic() { return "bsd-archive"; }

    static llvm::StringRef GetPluginDescriptionStatic() {
        return "BSD Archive object container reader.";
    }

    static lldb_private::ObjectContainer *
    CreateInstance(const lldb::ModuleSP &module_sp, lldb::DataBufferSP &data_sp,
                  lldb::offset_t data_offset, const lldb_private::FileSpec *file,
                  lldb::offset_t offset, lldb::offset_t length);

    static size_t GetModuleSpecifications(const
    lldb_private::FileSpec &file,
                                         lldb::DataBufferSP &data_sp,
                                         lldb::offset_t data_offset,

```

```

        lldb::offset_t file_offset,
        lldb::offset_t length,
        lldb_private::ModuleSpecList &specs);

static ArchiveType MagicBytesMatch(const lldb_private::DataExtractor &data);

// Member Functions
bool ParseHeader() override;

size_t GetNumObjects() const override {
    if (m_archive_sp)
        return m_archive_sp->GetNumObjects();
    return 0;
}

lldb::ObjectFileSP GetObjectFile(const lldb_private::FileSpec *file) override;

// PluginInterface protocol
llvm::StringRef GetPluginName() override { return GetPluginNameStatic(); }

protected:
struct Object {
    Object();

    void Clear();

    lldb::offset_t ExtractFromThin(const lldb_private::DataExtractor &data,

        lldb::offset_t offset,
        llvm::StringRef stringTable);

    lldb::offset_t Extract(const lldb_private::DataExtractor &data,
        lldb::offset_t offset);
    /// Object name in the archive.
    lldb_private::ConstString ar_name;

    /// Object modification time in the archive.
    uint32_t modification_time = 0;

    /// Object size in bytes in the archive.
    uint32_t size = 0;

    /// File offset in bytes from the beginning of the file of the object data.
    lldb::offset_t file_offset = 0;

    /// Length of the object data.
    lldb::offset_t file_size = 0;

```

```

void Dump() const;
};

class Archive {
public:
    typedef std::shared_ptr<Archive> shared_ptr;
    typedef std::multimap<lldb_private::FileSpec, shared_ptr> Map;

    Archive(const lldb_private::ArchSpec &arch,
            const llvm::sys::TimePoint<> &mod_time, lldb::offset_t file_offset,
            lldb_private::DataExtractor &data, ArchiveType archive_type);

    ~Archive();

    static Map &GetArchiveCache();

    static std::recursive_mutex &GetArchiveCacheMutex();

    static Archive::shared_ptr FindCachedArchive(
        const lldb_private::FileSpec &file, const lldb_private::ArchSpec &arch,
        const llvm::sys::TimePoint<> &mod_time, lldb::offset_t file_offset);

    static Archive::shared_ptr ParseAndCacheArchiveForFile(
        const lldb_private::FileSpec &file, const lldb_private::ArchSpec &arch,
        const llvm::sys::TimePoint<> &mod_time, lldb::offset_t file_offset,
        lldb_private::DataExtractor &data, ArchiveType archive_type);

    size_t GetNumObjects() const { return m_objects.size(); }

    const Object *GetObjectAtIndex(size_t idx) {
        if (idx < m_objects.size())
            return &m_objects[idx];
        return nullptr;
    }

    size_t ParseObjects();

    Object *FindObject(lldb_private::ConstString object_name,
                      const llvm::sys::TimePoint<> &object_mod_time);

    lldb::offset_t
    GetFileOffset() const { return m_file_offset; }

    const llvm::sys::TimePoint<> &GetModificationTime() {
        return m_modification_time;
    }

    const lldb_private::ArchSpec &GetArchitecture() const { return m_arch; }

```

```

void SetArchitecture(const lldb_private::ArchSpec &arch) { m_arch = arch; }

bool HasNoExternalReferences() const;

lldb_private::DataExtractor &GetData() { return m_data; }

ArchiveType GetArchiveType() { return m_archive_type; }

protected:
    typedef lldb_private::UniqueCStringMap<uint32_t> ObjectNameToIndexMap;
    // Member Variables
    lldb_private::ArchSpec m_arch;
    llvm::sys::TimePoint<> m_modification_time;
    lldb::offset_t m_file_offset;
    std::vector<Object> m_objects;
    ObjectNameToIndexMap m_object_name_to_index_map;
    lldb_private::DataExtractor m_data; ///< The data for this object container
                                     ///

```

```
    call sub()
!$omp end parallel
```

```
end
```

```
subroutine lastprivate()
class(*), allocatable, save :: x
```

!PORTABILITY: If a polymorphic variable with allocatable attribute 'x' is in LASTPRIVATE clause, the behavior is unspecified

```
!$omp do lastprivate(x)
do i = 1, 10
    call sub()
enddo
!$omp end do
```

```
end
```

```
subroutine copyin()
class(*), allocatable, save :: x
!$omp threadprivate(x)
```

!PORTABILITY: If a polymorphic variable with allocatable attribute 'x' is in COPYIN clause, the behavior is unspecified

```
!$omp parallel copyin(x)
    call sub()
!$omp end parallel
```

```
end
```

```
subroutine copyprivate()
class(*), allocatable, save :: x
!$omp threadprivate(x)
```

```
!$omp single
    call sub()
```

!PORTABILITY: If a polymorphic variable with allocatable attribute 'x' is in COPYPRIVATE clause, the behavior is unspecified

```
!$omp end single copyprivate(x)
```

```
end
```

```
=====
The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:
=====
```

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the



origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions  
of this Software are embedded into an Object form of such source code, you  
may redistribute such embedded portions in such Object form without complying  
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled  
forms of this Software with  
software that is licensed under the GPLv2 ("Combined Software") and if a  
court of competent jurisdiction determines that the patent provision (Section  
3), the indemnity provision (Section 9) or other Section of the License  
conflicts with the conditions of the GPLv2, you may retroactively and  
prospectively choose to deem waived or otherwise exclude such Section(s) of  
the License, but only in their entirety and only with respect to the Combined  
Software.

=====  
Software from third parties included in the LLVM Project:  
=====

The LLVM Project contains third party software which is under different license  
terms. All such code will be identified clearly using at least one of two  
mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or  
`LICENSE` file at the top containing the specific  
license and restrictions  
which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every  
file.

Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):

=====

The libunwind library is dual licensed under both the University of Illinois "BSD-Like" license and the MIT license. As a user of this code you may choose to use it under either license. As a contributor, you agree to allow your code to be used under both.

Full text of the relevant licenses is included below.

=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2009-2019 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

<http://llvm.org>

#### Permission

is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.

- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.

- \* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====  
Copyright (c) 2009-2014 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the

Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```
static_library("BSD-Archive") {
  output_name = "lldbPluginObjectContainerBSDArchive"
  configs += [ "../llvm/utils/gn/build:lldb_code" ]
  deps = [
    "../lldb/source/Core",
    "../lldb/source/Host",
    "../lldb/source/Symbol",
    "../llvm/lib/Support",
  ]
  sources = [ "ObjectContainerBSDArchive.cpp" ]
}
; RUN: opt -mtriple=aarch64-linux-gnu -mattr=+sve -passes=sccp,masked-gather-intrin -S < %s | FileCheck %s

; Testing that masked gathers operating on scalable vectors that are
; packed in SVE registers are not scalarized.

; CHECK-LABEL: @masked_gather_nxv4i32(
; CHECK: call <vscale x 4 x i32> @llvm.masked.gather.nxv4i32
```

```
define <vscale x 4 x i32> @masked_gather_nxv4i32(<vscale x 4 x ptr> %ld, <vscale x 4 x i1> %masks, <vscale x 4 x i32> %passthru) {
    %res = call <vscale x 4 x i32> @llvm.masked.gather.nxv4i32(<vscale x 4 x ptr> %ld, i32 0, <vscale x 4 x i1> %masks, <vscale x 4 x i32> %passthru)
    ret <vscale x 4 x i32> %res
}
```

; Testing that masked gathers operating on scalable vectors of FP data  
; that is packed in SVE registers are not scalarized.

```
; CHECK-LABEL: @masked_gather_nxv2f64(
; CHECK: call <vscale x 2 x double> @llvm.masked.gather.nxv2f64
define <vscale x 2 x double> @masked_gather_nxv2f64(<vscale x 2 x ptr> %ld, <vscale x 2 x i1> %masks, <vscale x 2 x double> %passthru)
{
    %res = call <vscale x 2 x double> @llvm.masked.gather.nxv2f64(<vscale x 2 x ptr> %ld, i32 0, <vscale x 2 x i1> %masks, <vscale x 2 x double> %passthru)
    ret <vscale x 2 x double> %res
}
```

; Testing that masked gathers operating on scalable vectors of FP data  
; that is unpacked in SVE registers are not scalarized.

```
; CHECK-LABEL: @masked_gather_nxv2f16(
; CHECK: call <vscale x 2 x half> @llvm.masked.gather.nxv2f16
define <vscale x 2 x half> @masked_gather_nxv2f16(<vscale x 2 x ptr> %ld, <vscale x 2 x i1> %masks, <vscale x 2 x half> %passthru) {
    %res = call <vscale x 2 x half> @llvm.masked.gather.nxv2f16(<vscale x 2 x ptr> %ld, i32 0, <vscale x 2 x i1> %masks, <vscale x 2 x half> %passthru)
    ret <vscale x 2 x half> %res
}
```

; Testing that masked gathers operating on 64-bit fixed vectors are  
; scalarized because NEON doesn't have support for masked gather  
; instructions.

```
; CHECK-LABEL: @masked_gather_v2f32(
; CHECK-NOT: @llvm.masked.gather.v2f32(
define <2 x float> @masked_gather_v2f32(<2 x ptr> %ld, <2 x i1> %masks, <2 x float> %passthru) {
    %res = call <2 x float> @llvm.masked.gather.v2f32(<2 x ptr> %ld, i32 0, <2 x i1> %masks, <2 x float> %passthru)
    ret <2 x float> %res
}
```

; Testing that masked gathers operating on 128-bit fixed vectors are  
; scalarized because NEON doesn't have support for masked gather  
; instructions and because we are not targeting fixed width SVE.

```
; CHECK-LABEL: @masked_gather_v4i32(
; CHECK-NOT: @llvm.masked.gather.v4i32(
define <4 x i32> @masked_gather_v4i32(<4 x ptr> %ld, <4 x i1> %masks, <4 x i32> %passthru) {
    %res = call <4 x i32> @llvm.masked.gather.v4i32(<4 x ptr> %ld, i32 0, <4 x i1> %masks, <4 x i32> %passthru)
    ret <4 x i32> %res
}

declare <vscale x 4 x i32> @llvm.masked.gather.nxv4i32(<vscale x 4 x ptr> %ptrs, i32 %align, <vscale x 4 x i1>
%.masks, <vscale x 4 x i32> %passthru)
declare <vscale x 2 x double> @llvm.masked.gather.nxv2f64(<vscale x 2 x ptr> %ptrs, i32 %align, <vscale x 2 x
i1> %masks, <vscale x 2 x double>
%passthru)
declare <vscale x 2 x half> @llvm.masked.gather.nxv2f16(<vscale x 2 x ptr> %ptrs, i32 %align, <vscale x 2 x i1>
%.masks, <vscale x 2 x half> %passthru)
declare <2 x float> @llvm.masked.gather.v2f32(<2 x ptr> %ptrs, i32 %align, <2 x i1> %masks, <2 x float>
%passthru)
declare <4 x i32> @llvm.masked.gather.v4i32(<4 x ptr> %ptrs, i32 %align, <4 x i1> %masks, <4 x i32> %passthru)
# RUN: llc -O0 -mtriple=m68k -start-after=prologuepillog -verify-machineinstrs %s -o - | FileCheck %s
```

name: is-pcrel-register-operand-legal

body: |

bb.0.entry:

```
; CHECK: move.l (0,%pc,%a0), (%a1)
```

```
; CHECK: move.l (%a0), (0,%pc,%a1)
```

```
MOV32jk $a1, 0, $a0, implicit-def $ccr
```

```
MOV32kj 0, $a1, $a0, implicit-def $ccr
```

Ptyprocess is under the ISC license, as code derived from Pexpect.

<http://opensource.org/licenses/ISC>

Copyright (c) 2013-2014, Pexpect development team

Copyright (c) 2012, Noah Spurrier <noah@noah.org>

PERMISSION TO USE, COPY, MODIFY, AND/OR DISTRIBUTE THIS SOFTWARE FOR ANY PURPOSE WITH OR WITHOUT FEE IS HEREBY GRANTED, PROVIDED THAT THE ABOVE COPYRIGHT NOTICE AND THIS PERMISSION NOTICE APPEAR IN ALL COPIES. THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

UNRECOGNIZED LICENSE; MD5 sum: 2cc784086c8a087e4008cc9f558624bd

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the

power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition,

"submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:



- (a) You must give any other recipients of the Work or  
Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices  
stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works  
that You distribute, all copyright, patent, trademark, and  
attribution notices from the Source form of the Work,  
excluding those notices that do not pertain to any part of  
the Derivative Works; and
- (d) If the Work  
includes a "NOTICE" text file as part of its  
distribution, then any Derivative Works that You distribute must  
include a readable copy of the attribution notices contained  
within such NOTICE file, excluding those notices that do not  
pertain to any part of the Derivative Works, in at least one  
of the following places: within a NOTICE text file distributed  
as part of the Derivative Works; within the Source form or  
documentation, if provided along with the Derivative Works; or,  
within a display generated by the Derivative Works, if and  
wherever such third-party notices normally appear. The contents  
of the NOTICE file are for informational purposes only and  
do not modify the License. You may add Your own attribution  
notices within Derivative Works that You distribute, alongside  
or as an addendum to the NOTICE text from the Work, provided  
that such additional  
attribution notices cannot be construed  
as modifying the License.

You may add Your own copyright statement to Your modifications and  
may provide additional or different license terms and conditions  
for use, reproduction, or distribution of Your modifications, or  
for any such Derivative Works as a whole, provided Your use,  
reproduction, and distribution of the Work otherwise complies with  
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,  
any Contribution intentionally submitted for inclusion in the Work  
by You to the Licensor shall be under the terms and conditions of  
this License, without any additional terms or conditions.  
Notwithstanding the above, nothing herein shall supersede or modify  
the terms of any separate license agreement you may have executed  
with Licensor regarding such Contributions.

6. Trademarks. This License

does not grant permission to use the trade

names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate

comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

#### ---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

=====  
Software from third parties included in the LLVM Project:  
=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or `LICENSE` file at the top containing the specific license and restrictions which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every

file.

# People who have agreed to one of the CLAs and can contribute patches.

# The AUTHORS file lists the copyright holders; this file

# lists people. For example, Google employees are listed here

# but not in AUTHORS, because Google holds the copyright.

#

# Names should be added to this file only after verifying that

# the individual or the individual's organization has agreed to

# the appropriate Contributor License Agreement, found here:

#

# <https://developers.google.com/open-source/cla/individual>

# <https://developers.google.com/open-source/cla/corporate>

#

# The agreement for individuals can be filled out on the web.

#

# When adding J Random Contributor's name to this file,

# either J's name or J's organization's name should be

# added to the AUTHORS file, depending on whether the

# individual or corporate CLA was used.

#

# Names should be added to this file as:

# Name <email address>

#

# Please keep the list sorted.

Abhina Sreeskantharajan <abhina.sreeskantharajan@ibm.com>

Albert Pretorius

<pretoalb@gmail.com>

Alex Steele <steelal123@gmail.com>

Andriy Berestovskyy <berestovskyy@gmail.com>

Arne Beer <arne@twobeer.de>

Billy Robert O'Neal III <billy.oneal@gmail.com> <bion@microsoft.com>

Chris Kennelly <ckennelly@google.com> <ckennelly@ckennelly.com>

Christian Wassermann <christian\_wassermann@web.de>

Christopher Seymour <chris.j.seymour@hotmail.com>

Colin Braley <braley.colin@gmail.com>

Cyrille Faucheux <cyrille.faucheux@gmail.com>

Daniel Harvey <danielharvey458@gmail.com>

David Coeurjolly <david.coeurjolly@liris.cnrs.fr>

Deniz Evrenci <denizevrenci@gmail.com>

Dominic Hamon <dma@stripsock.com> <dominic@google.com>

Dominik Czarnota <dominik.b.czarnota@gmail.com>

Dominik Korman <kormandominik@gmail.com>

Donald Aingworth <donalds\_junk\_mail@yahoo.com>

Eric Backus <eric\_backus@alum.mit.edu>

Eric Fiselier <eric@efcs.ca>

Eugene Zhuk <eugene.zhuk@gmail.com>

Evgeny Safronov <division494@gmail.com>

Fanbo Meng <fanbo.meng@ibm.com>  
Federico Ficarelli <federico.ficarelli@gmail.com>  
Felix  
Homann <linuxaudio@showlabor.de>  
Geoffrey Martin-Noble <gcmn@google.com> <gmngeoffrey@gmail.com>  
Gerg Szitr <szitar.gergo@gmail.com>  
Hannes Hauswedell <h2@fsfe.org>  
Ismael Jimenez Martinez <ismael.jimenez.martinez@gmail.com>  
Jern-Kuan Leong <jernkuan@gmail.com>  
JianXiong Zhou <zhoujianxiong2@gmail.com>  
Joao Paulo Magalhaes <joaoppmagalhaes@gmail.com>  
John Millikin <jmillikin@stripe.com>  
Jordan Williams <jwillikers@protonmail.com>  
Jussi Knuuttila <jussi.knuuttila@gmail.com>  
Kai Wolf <kai.wolf@gmail.com>  
Kaito Udagawa <umireon@gmail.com>  
Kishan Kumar <kumar.kishan@outlook.com>  
Lei Xu <eddyxu@gmail.com>  
Matt Clarkson <mattyclarkson@gmail.com>  
Maxim Vafin <maxvafin@gmail.com>  
Nick Hutchinson <nshutchinson@gmail.com>  
Norman Heino <norman.heino@gmail.com>  
Oleksandr Sochka <sasha.sochka@gmail.com>  
Ori Livneh <ori.livneh@gmail.com>  
Pascal Leroy <phl@google.com>  
Paul Redmond <paul.redmond@gmail.com>  
Pierre Phaneuf <pphaneuf@google.com>  
Radoslav Yovchev <radoslav.tm@gmail.com>  
Raul Marin  
<rmrodriguez@cartodb.com>  
Ray Glover <ray.glover@uk.ibm.com>  
Robert Guo <robert.guo@mongodb.com>  
Roman Lebedev <lebedev.ri@gmail.com>  
Sayan Bhattacharjee <aero.sayan@gmail.com>  
Shuo Chen <chenshuo@chenshuo.com>  
Steven Wan <wan.yu@ibm.com>  
Tobias Schmidt <tobias.schmidt@in.tum.de>  
Tobias Ulvgrd <tobias.ulvgard@dirac.se>  
Tom Madams <tom.ej.madams@gmail.com> <tmadams@google.com>  
Yixuan Qiu <yixuanq@gmail.com>  
Yusuke Suzuki <utatane.tea@gmail.com>  
Zbigniew Skowron <zbychs@gmail.com>  
Min-Yih Hsu <yihshyng223@gmail.com>

=====  
The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:  
=====

Apache License

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the

power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition,

"submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or  
Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices  
stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works  
that You distribute, all copyright, patent, trademark, and  
attribution notices from the Source form of the Work,  
excluding those notices that do not pertain to any part of  
the Derivative Works; and
- (d) If the Work  
includes a "NOTICE" text file as part of its  
distribution, then any Derivative Works that You distribute must  
include a readable copy of the attribution notices contained  
within such NOTICE file, excluding those notices that do not  
pertain to any part of the Derivative Works, in at least one  
of the following places: within a NOTICE text file distributed  
as part of the Derivative Works; within the Source form or  
documentation, if provided along with the Derivative Works; or,  
within a display generated by the Derivative Works, if and  
wherever such third-party notices normally appear. The contents  
of the NOTICE file are for informational purposes only and  
do not modify the License. You may add Your own attribution  
notices within Derivative Works that You distribute, alongside  
or as an addendum to the NOTICE text from the Work, provided  
that such additional  
attribution notices cannot be construed  
as modifying the License.

You may add Your own copyright statement to Your modifications and  
may provide additional or different license terms and conditions  
for use, reproduction, or distribution of Your modifications, or  
for any such Derivative Works as a whole, provided Your use,  
reproduction, and distribution of the Work otherwise complies with  
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,  
any Contribution intentionally submitted for inclusion in the Work  
by You to the Licensor shall be under the terms and conditions of  
this License, without any additional terms or conditions.  
Notwithstanding the above, nothing herein shall supersede or modify  
the terms of any separate license agreement you may have executed  
with Licensor regarding such Contributions.

6. Trademarks. This License



does not grant permission to use the trade

names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate

comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

#### ---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

=====  
Software from third parties included in the LLVM Project:  
=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or `LICENSE` file at the top containing the specific license and restrictions which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every

file.

=====

Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):

=====

University of Illinois/NCSA

Open Source License

Copyright (c) 2007-2018 University of Illinois at Urbana-Champaign.

All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

<http://llvm.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.

- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.

- \* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

; NOTE: Assertions have been autogenerated by utils/update\_llc\_test\_checks.py

; RUN: llc < %s -mtriple=x86\_64-linux-android -mattr=+mmx -enable-legalize-types-checking | FileCheck %s

;

; D31946

; Check that we dont end up with the ""LLVM ERROR: Cannot select" error.

; Additionally ensure that the output code actually put fp128 values in SSE registers.

```

declare fp128 @llvm.fabs.f128(fp128)
declare fp128 @llvm.copysign.f128(fp128, fp128)

define fp128 @TestSelect(fp128 %a, fp128 %b) {
; CHECK-LABEL: TestSelect:
; CHECK:      # %bb.0:
; CHECK-NEXT: pushq %rbx
; CHECK-NEXT: .cfi_def_cfa_offset 16
; CHECK-NEXT: subq $32, %rsp
; CHECK-NEXT: .cfi_def_cfa_offset 48
; CHECK-NEXT: .cfi_offset %rbx, -16
; CHECK-NEXT: movaps %xmm1, {[[-0-9]+]}(%r{[sb]})p) # 16-byte Spill
; CHECK-NEXT: movaps %xmm0, (%rsp) # 16-byte Spill
; CHECK-NEXT: callq __gttf2@PLT
; CHECK-NEXT: movl %eax, %ebx
; CHECK-NEXT: movaps (%rsp), %xmm0 # 16-byte Reload
; CHECK-NEXT: movaps {[[-0-9]+]}(%r{[sb]})p),
    %xmm1 # 16-byte Reload
; CHECK-NEXT: callq __subtf3@PLT
; CHECK-NEXT: testl %ebx, %ebx
; CHECK-NEXT: jg .LBB0_2
; CHECK-NEXT: # %bb.1:
; CHECK-NEXT: xorps %xmm0, %xmm0
; CHECK-NEXT: .LBB0_2:
; CHECK-NEXT: addq $32, %rsp
; CHECK-NEXT: .cfi_def_cfa_offset 16
; CHECK-NEXT: popq %rbx
; CHECK-NEXT: .cfi_def_cfa_offset 8
; CHECK-NEXT: retq
    %cmp = fcmp ogt fp128 %a, %b
    %sub = fsub fp128 %a, %b
    %res = select i1 %cmp, fp128 %sub, fp128 0xL00000000000000000000000000000000
    ret fp128 %res
}

define fp128 @TestFabs(fp128 %a) {
; CHECK-LABEL: TestFabs:
; CHECK:      # %bb.0:
; CHECK-NEXT: andps {[.?.LCPI[0-9]+_[0-9]+]}(%rip), %xmm0
; CHECK-NEXT: retq
    %res = call fp128 @llvm.fabs.f128(fp128 %a)
    ret fp128 %res
}

define fp128 @TestCopysign(fp128 %a, fp128 %b) {
; CHECK-LABEL: TestCopysign:
; CHECK:      # %bb.0:

```

```

; CHECK-NEXT: andps {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %xmm1
; CHECK-NEXT: andps {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %xmm0
; CHECK-NEXT:
    orps %xmm1, %xmm0
; CHECK-NEXT: retq
%res = call fp128 @llvm.copysign.f128(fp128 %a, fp128 %b)
ret fp128 %res
}

```

```

define fp128 @TestFneg(fp128 %a) {
; CHECK-LABEL: TestFneg:
; CHECK:      # %bb.0:
; CHECK-NEXT: pushq %rax
; CHECK-NEXT: .cfi_def_cfa_offset 16
; CHECK-NEXT: movaps %xmm0, %xmm1
; CHECK-NEXT: callq __multf3@PLT
; CHECK-NEXT: xorps {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %xmm0
; CHECK-NEXT: popq %rax
; CHECK-NEXT: .cfi_def_cfa_offset 8
; CHECK-NEXT: retq
%mul = fmul fp128 %a, %a
%res = fsub fp128 0xL00000000000000008000000000000000, %mul
ret fp128 %res
}

```

-----

clang-tidy CERT Files

-----

All clang-tidy files are licensed under the same terms as the rest of the LLVM project with the following additions:

Any file referencing a CERT Secure Coding guideline:

Please allow this letter to serve as confirmation that open source projects on <http://llvm.org> are permitted to link via hypertext to the CERT(R) secure coding guidelines available at <https://www.securecoding.cert.org>.

The foregoing is permitted by the Terms of Use as follows:

"Linking to the Service

Because we update many of our Web documents regularly, we would prefer that you link to our Web pages whenever possible rather than reproduce them. It is not necessary to request permission to make referential hypertext links to The Service."

<http://www.sei.cmu.edu/legal/ip/index.cfm>.

Please allow this letter to also confirm that no formal permission is required

to reproduce the title of the content being linked to, nor to reproduce any de Minimis description of such content.

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the

power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition,

"submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work

includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional

attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify



the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

#### ---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

=====  
Software from third parties included in the LLVM Project:  
=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

1) It will be in a separate directory tree with its own `LICENSE.txt` or

`LICENSE` file at the top containing the specific  
license and restrictions  
which apply to that software, or  
2) It will contain specific license and restriction terms at the top of every  
file.

=====

Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):

=====

The libclc library is dual licensed under both the University of Illinois  
"BSD-Like" license and the MIT license. As a user of this code you may choose  
to use it under either license. As a contributor, you agree to allow your code  
to be used under both.

Full text of the relevant licenses is included below.

=====

Copyright (c) 2011-2019 by the contributors listed in CREDITS.TXT

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of  
this software and associated documentation files (the "Software"),  
to deal with  
the Software without restriction, including without limitation the rights to  
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies  
of the Software, and to permit persons to whom the Software is furnished to do  
so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice,  
this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice,  
this list of conditions and the following disclaimers in the  
documentation and/or other materials provided with the distribution.
- \* The names of the contributors may not be used to endorse or promote  
products derived from this Software without specific prior written  
permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS  
FOR A PARTICULAR  
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE  
CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====

Copyright (c) 2011-2014 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE

SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

; NOTE: Assertions have been autogenerated by utils/update\_llc\_test\_checks.py

; RUN: llc -mtriple=riscv64 -mattr=+zbbk -verify-machineinstrs < %s \

; RUN: | FileCheck %s -check-prefix=RV64ZBKB

declare i64 @llvm.riscv.brev8.i64(i64)

define i64 @brev8(i64 %a) nounwind {

; RV64ZBKB-LABEL: brev8:

; RV64ZBKB: # %bb.0:

; RV64ZBKB-NEXT: brev8 a0, a0

; RV64ZBKB-NEXT: ret

%val = call i64 @llvm.riscv.brev8.i64(i64 %a)

ret i64 %val

}

; Test that rev8 is recognized as preserving zero extension.

define zeroext i16 @brev8\_knownbits(i16 zeroext %a) nounwind {

; RV64ZBKB-LABEL: brev8\_knownbits:

; RV64ZBKB: # %bb.0:

; RV64ZBKB-NEXT: brev8 a0, a0

; RV64ZBKB-NEXT: ret

%zext = zext i16 %a to i64

%val = call i64 @llvm.riscv.brev8.i64(i64 %zext)

```

%trunc = trunc i64 %val to i16
ret i16 %trunc
}

declare i64 @llvm.bswap.i64(i64)

define i64 @rev8_i64(i64 %a) {
; RV64ZBKB-LABEL: rev8_i64:
; RV64ZBKB:      # %bb.0:
; RV64ZBKB-NEXT:  rev8 a0, a0
; RV64ZBKB-NEXT:  ret
    %1 = call i64 @llvm.bswap.i64(i64 %a)
ret i64 %1
}

declare i32 @llvm.riscv.brev8.i32(i32)

define signext i32 @brev8_i32(i32 signext %a) nounwind {
; RV64ZBKB-LABEL: brev8_i32:
; RV64ZBKB:      # %bb.0:
; RV64ZBKB-NEXT:  brev8 a0, a0
; RV64ZBKB-NEXT:  sext.w a0, a0
; RV64ZBKB-NEXT:  ret
    %val = call i32 @llvm.riscv.brev8.i32(i32 %a)
ret i32 %val
}

; Test that rev8 is recognized as preserving zero extension.
define zeroext i16 @brev8_i32_knownbits(i16 zeroext %a) nounwind {
; RV64ZBKB-LABEL: brev8_i32_knownbits:
; RV64ZBKB:      # %bb.0:
; RV64ZBKB-NEXT:  brev8 a0, a0
; RV64ZBKB-NEXT:  ret
    %zext = zext i16 %a to i32
    %val = call i32 @llvm.riscv.brev8.i32(i32 %zext)
    %trunc = trunc i32 %val to i16
ret i16 %trunc
}

declare i32 @llvm.bswap.i32(i32)

define signext i32 @rev8_i32(i32 signext %a) {
; RV64ZBKB-LABEL: rev8_i32:
; RV64ZBKB:      # %bb.0:
; RV64ZBKB-NEXT:  rev8 a0, a0
; RV64ZBKB-NEXT:  srai a0, a0, 32
; RV64ZBKB-NEXT:  ret
    %1 = call i32 @llvm.bswap.i32(i32

```

```

%a)
ret i32 %1
}
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=amdgc -mcpu=fiji < %s | FileCheck -check-prefixes=VI %s

; FIXME: This one should fold to rcp
define half @select_fneg_posk_src_rcp_f16(i32 %c, half %x, half %y) {
; VI-LABEL: select_fneg_posk_src_rcp_f16:
; VI:      ; %bb.0:
; VI-NEXT:  s_waitcnt vmcnt(0) expcnt(0) lgkmcnt(0)
; VI-NEXT:  v_rcp_f16_e64 v1, -v1
; VI-NEXT:  v_mov_b32_e32 v2, 0x4000
; VI-NEXT:  v_cmp_eq_u32_e32 vcc, 0, v0
; VI-NEXT:  v_cndmask_b32_e32 v0, v2, v1, vcc
; VI-NEXT:  s_setpc_b64 s[30:31]
%cmp = icmp eq i32 %c, 0
%rcp = call half @llvm.amdgc.rcp.f16(half %x)
%fneg = fneg half %rcp
%select = select i1 %cmp, half %fneg, half 2.0
ret half %select
}

declare half @llvm.amdgc.rcp.f16(half) #0

attributes #0 = { nocallback nofree nosync nounwind speculatable willreturn memory(none) }
Apache License 2.0
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefixes=CHECK,RV64I
; RUN: llc -mtriple=riscv64 -mattr=+zbb -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefixes=CHECK,RV64ZBB-ZBKB,RV64ZBB
; RUN: llc -mtriple=riscv64 -mattr=+zbkb -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefixes=CHECK,RV64ZBB-
ZBKB,RV64ZBKB

define signext i32 @andn_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: andn_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  not a1, a1
; RV64I-NEXT:  and a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: andn_i32:
; RV64ZBB-ZBKB:      # %bb.0:
; RV64ZBB-ZBKB-NEXT:  andn a0, a0, a1
; RV64ZBB-ZBKB-NEXT:  ret
%neg = xor i32 %b, -1

```

```

%and = and i32 %neg, %a
ret i32 %and
}

```

```

define i64 @andn_i64(i64 %a, i64
%b) nounwind {
; RV64I-LABEL: andn_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT: not a1, a1
; RV64I-NEXT: and a0, a1, a0
; RV64I-NEXT: ret
;
; RV64ZBB-ZBKB-LABEL: andn_i64:
; RV64ZBB-ZBKB:  # %bb.0:
; RV64ZBB-ZBKB-NEXT: andn a0, a0, a1
; RV64ZBB-ZBKB-NEXT: ret
%neg = xor i64 %b, -1
%and = and i64 %neg, %a
ret i64 %and
}

```

```

define signext i32 @orn_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: orn_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT: not a1, a1
; RV64I-NEXT: or a0, a1, a0
; RV64I-NEXT: ret
;
; RV64ZBB-ZBKB-LABEL: orn_i32:
; RV64ZBB-ZBKB:  # %bb.0:
; RV64ZBB-ZBKB-NEXT: orn a0, a0, a1
; RV64ZBB-ZBKB-NEXT: ret
%neg = xor i32 %b, -1
%or = or i32 %neg, %a
ret i32 %or
}

```

```

define i64 @orn_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: orn_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT: not a1, a1
; RV64I-NEXT: or a0, a1, a0
; RV64I-NEXT: ret
;
; RV64ZBB-ZBKB-LABEL: orn_i64:
; RV64ZBB-ZBKB:  # %bb.0:
; RV64ZBB-ZBKB-NEXT:

```

```

        orn a0, a0, a1
; RV64ZBB-ZBKB-NEXT:  ret
        %neg = xor i64 %b, -1
        %or = or i64 %neg, %a
        ret i64 %or
    }

define signext i32 @xnor_i32(i32 signext %a, i32 signext %b) nounwind {
; CHECK-LABEL: xnor_i32:
; CHECK:      # %bb.0:
; CHECK-NEXT:  xor a0, a0, a1
; CHECK-NEXT:  not a0, a0
; CHECK-NEXT:  ret
        %neg = xor i32 %a, -1
        %xor = xor i32 %neg, %b
        ret i32 %xor
    }

define i64 @xnor_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: xnor_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: xnor_i64:
; RV64ZBB-ZBKB:  # %bb.0:
; RV64ZBB-ZBKB-NEXT:  xnor a0, a0, a1
; RV64ZBB-ZBKB-NEXT:  ret
        %neg = xor i64 %a, -1
        %xor = xor i64 %neg, %b
        ret i64 %xor
    }

declare i32 @llvm.fshl.i32(i32, i32, i32)

define signext i32 @rol_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: rol_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  andi a2, a1, -1
; RV64I-NEXT:  sllw a1, a0, a1
; RV64I-NEXT:
        negw a2, a2
; RV64I-NEXT:  srlw a0, a0, a2
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: rol_i32:

```



```

; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  rolw a0, a0, a1
; RV64ZBB-ZBKB-NEXT:  ret
%1 = tail call i32 @llvm.fshl.i32(i32 %a, i32 %a, i32 %b)
ret i32 %1
}

; Similar to rol_i32, but doesn't sign extend the result.
define void @rol_i32_nosext(i32 signext %a, i32 signext %b, ptr %x) nounwind {
; RV64I-LABEL: rol_i32_nosext:
; RV64I:    # %bb.0:
; RV64I-NEXT:  andi a3, a1, -1
; RV64I-NEXT:  sllw a1, a0, a1
; RV64I-NEXT:  negw a3, a3
; RV64I-NEXT:  srlw a0, a0, a3
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:  sw a0, 0(a2)
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: rol_i32_nosext:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  rolw a0, a0, a1
; RV64ZBB-ZBKB-NEXT:  sw a0, 0(a2)
; RV64ZBB-ZBKB-NEXT:  ret
%1 = tail call i32 @llvm.fshl.i32(i32 %a, i32 %a, i32 %b)
store i32 %1, ptr %x
ret void
}

define signext
i32 @rol_i32_neg_constant_rhs(i32 signext %a) nounwind {
; RV64I-LABEL: rol_i32_neg_constant_rhs:
; RV64I:    # %bb.0:
; RV64I-NEXT:  andi a1, a0, -1
; RV64I-NEXT:  li a2, -2
; RV64I-NEXT:  sllw a0, a2, a0
; RV64I-NEXT:  negw a1, a1
; RV64I-NEXT:  srlw a1, a2, a1
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: rol_i32_neg_constant_rhs:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  li a1, -2
; RV64ZBB-ZBKB-NEXT:  rolw a0, a1, a0
; RV64ZBB-ZBKB-NEXT:  ret
%1 = tail call i32 @llvm.fshl.i32(i32 -2, i32 -2, i32 %a)
ret i32 %1

```

```
}
```

```
declare i64 @llvm.fshl.i64(i64, i64, i64)
```

```
define i64 @rol_i64(i64 %a, i64 %b) nounwind {  
; RV64I-LABEL: rol_i64:  
; RV64I:      # %bb.0:  
; RV64I-NEXT: sll a2, a0, a1  
; RV64I-NEXT: negw a1, a1  
; RV64I-NEXT: srl a0, a0, a1  
; RV64I-NEXT: or a0, a2, a0  
; RV64I-NEXT: ret  
;  
; RV64ZBB-ZBKB-LABEL: rol_i64:  
; RV64ZBB-ZBKB:  # %bb.0:  
; RV64ZBB-ZBKB-NEXT:  rol a0, a0, a1  
; RV64ZBB-ZBKB-NEXT:  
    ret  
%or = tail call i64 @llvm.fshl.i64(i64 %a, i64 %a, i64 %b)  
ret i64 %or  
}
```

```
declare i32 @llvm.fshr.i32(i32, i32, i32)
```

```
define signext i32 @ror_i32(i32 signext %a, i32 signext %b) nounwind {  
; RV64I-LABEL: ror_i32:  
; RV64I:      # %bb.0:  
; RV64I-NEXT: andi a2, a1, -1  
; RV64I-NEXT: srlw a1, a0, a1  
; RV64I-NEXT: negw a2, a2  
; RV64I-NEXT: sllw a0, a0, a2  
; RV64I-NEXT: or a0, a1, a0  
; RV64I-NEXT: ret  
;  
; RV64ZBB-ZBKB-LABEL: ror_i32:  
; RV64ZBB-ZBKB:  # %bb.0:  
; RV64ZBB-ZBKB-NEXT:  rorw a0, a0, a1  
; RV64ZBB-ZBKB-NEXT:  ret  
%1 = tail call i32 @llvm.fshr.i32(i32 %a, i32 %a, i32 %b)  
ret i32 %1  
}
```

; Similar to ror\_i32, but doesn't sign extend the result.

```
define void @ror_i32_nosext(i32 signext %a, i32 signext %b, ptr %x) nounwind {  
; RV64I-LABEL: ror_i32_nosext:  
; RV64I:      # %bb.0:  
; RV64I-NEXT: andi a3, a1, -1  
; RV64I-NEXT: srlw a1, a0, a1
```

```

; RV64I-NEXT:  negw a3, a3
; RV64I-NEXT:  sllw a0, a0, a3
; RV64I-NEXT:  or a0, a1, a0
; RV64I-NEXT:
    sw a0, 0(a2)
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: ror_i32_nosext:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  rorw a0, a0, a1
; RV64ZBB-ZBKB-NEXT:  sw a0, 0(a2)
; RV64ZBB-ZBKB-NEXT:  ret
%1 = tail call i32 @llvm.fshr.i32(i32 %a, i32 %a, i32 %b)
store i32 %1, ptr %x
ret void
}

```

```

define signext i32 @ror_i32_neg_constant_rhs(i32 signext %a) nounwind {
; RV64I-LABEL: ror_i32_neg_constant_rhs:
; RV64I:    # %bb.0:
; RV64I-NEXT:  andi a1, a0, -1
; RV64I-NEXT:  li a2, -2
; RV64I-NEXT:  srlw a0, a2, a0
; RV64I-NEXT:  negw a1, a1
; RV64I-NEXT:  sllw a1, a2, a1
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: ror_i32_neg_constant_rhs:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  li a1, -2
; RV64ZBB-ZBKB-NEXT:  rorw a0, a1, a0
; RV64ZBB-ZBKB-NEXT:  ret
%1 = tail call i32 @llvm.fshr.i32(i32 -2, i32 -2, i32 %a)
ret i32 %1
}

```

```

declare i64 @llvm.fshr.i64(i64, i64, i64)

```

```

define i64 @ror_i64(i64 %a, i64 %b) nounwind {
;
RV64I-LABEL: ror_i64:
; RV64I:    # %bb.0:
; RV64I-NEXT:  srl a2, a0, a1
; RV64I-NEXT:  negw a1, a1
; RV64I-NEXT:  sll a0, a0, a1
; RV64I-NEXT:  or a0, a2, a0
; RV64I-NEXT:  ret

```

```

;
; RV64ZBB-ZBKB-LABEL: ror_i64:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  ror a0, a0, a1
; RV64ZBB-ZBKB-NEXT:  ret
%or = tail call i64 @llvm.fshr.i64(i64 %a, i64 %a, i64 %b)
ret i64 %or
}

define signext i32 @rori_i32_fshl(i32 signext %a) nounwind {
; RV64I-LABEL: rori_i32_fshl:
; RV64I:    # %bb.0:
; RV64I-NEXT:  srlw a1, a0, 1
; RV64I-NEXT:  sllw a0, a0, 31
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: rori_i32_fshl:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  roriw a0, a0, 1
; RV64ZBB-ZBKB-NEXT:  ret
%1 = tail call i32 @llvm.fshl.i32(i32 %a, i32 %a, i32 31)
ret i32 %1
}

; Similar to rori_i32_fshl, but doesn't sign extend the result.
define void @rori_i32_fshl_nosext(i32 signext %a, ptr %x) nounwind {
; RV64I-LABEL: rori_i32_fshl_nosext:
;
RV64I:    # %bb.0:
; RV64I-NEXT:  srlw a2, a0, 1
; RV64I-NEXT:  slli a0, a0, 31
; RV64I-NEXT:  or a0, a0, a2
; RV64I-NEXT:  sw a0, 0(a1)
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: rori_i32_fshl_nosext:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  roriw a0, a0, 1
; RV64ZBB-ZBKB-NEXT:  sw a0, 0(a1)
; RV64ZBB-ZBKB-NEXT:  ret
%1 = tail call i32 @llvm.fshl.i32(i32 %a, i32 %a, i32 31)
store i32 %1, ptr %x
ret void
}

define signext i32 @rori_i32_fshr(i32 signext %a) nounwind {
; RV64I-LABEL: rori_i32_fshr:

```

```

; RV64I:    # %bb.0:
; RV64I-NEXT:  slliw a1, a0, 1
; RV64I-NEXT:  srliw a0, a0, 31
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: rori_i32_fshr:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  roriw a0, a0, 31
; RV64ZBB-ZBKB-NEXT:  ret
%1 = tail call i32 @llvm.fshr.i32(i32 %a, i32 %a, i32 31)
ret i32 %1
}

```

; Similar to rori\_i32\_fshr, but doesn't sign extend the result.

```

define void @rori_i32_fshr_nosext(i32
  signext %a, ptr %x) nounwind {
; RV64I-LABEL: rori_i32_fshr_nosext:
; RV64I:    # %bb.0:
; RV64I-NEXT:  slli a2, a0, 1
; RV64I-NEXT:  srliw a0, a0, 31
; RV64I-NEXT:  or a0, a0, a2
; RV64I-NEXT:  sw a0, 0(a1)
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: rori_i32_fshr_nosext:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  roriw a0, a0, 31
; RV64ZBB-ZBKB-NEXT:  sw a0, 0(a1)
; RV64ZBB-ZBKB-NEXT:  ret
%1 = tail call i32 @llvm.fshr.i32(i32 %a, i32 %a, i32 31)
store i32 %1, ptr %x
ret void
}

```

; This test is similar to the type legalized version of the fshl/fshr tests, but

; instead of having the same input to both shifts it has different inputs. Make

; sure we don't match it as a roriw.

```

define signext i32 @not_rori_i32(i32 signext %x, i32 signext %y) nounwind {
; CHECK-LABEL: not_rori_i32:
; CHECK:    # %bb.0:
; CHECK-NEXT:  slliw a0, a0, 31
; CHECK-NEXT:  srliw a1, a1, 1
; CHECK-NEXT:  or a0, a0, a1
; CHECK-NEXT:  ret
%a = shl
i32 %x, 31
%b = lshr i32 %y, 1

```

```

%c = or i32 %a, %b
ret i32 %c
}

```

; This is similar to the type legalized roriw pattern, but the and mask is more  
; than 32 bits so the lshr doesn't shift zeroes into the lower 32 bits. Make  
; sure we don't match it to roriw.

```

define i64 @roriw_bug(i64 %x) nounwind {
; CHECK-LABEL: roriw_bug:
; CHECK:      # %bb.0:
; CHECK-NEXT:  slli a1, a0, 31
; CHECK-NEXT:  andi a2, a0, -2
; CHECK-NEXT:  srli a0, a0, 1
; CHECK-NEXT:  or a0, a1, a0
; CHECK-NEXT:  sext.w a0, a0
; CHECK-NEXT:  xor a0, a2, a0
; CHECK-NEXT:  ret
%a = shl i64 %x, 31
%b = and i64 %x, 18446744073709551614
%c = lshr i64 %b, 1
%d = or i64 %a, %c
%e = shl i64 %d, 32
%f = ashr i64 %e, 32
%g = xor i64 %b, %f ; to increase the use count on %b to disable SimplifyDandedBits.
ret i64 %g
}

```

```

define i64 @rori_i64_fshl(i64 %a) nounwind {
; RV64I-LABEL: rori_i64_fshl:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srli a1, a0, 1
; RV64I-NEXT:  slli a0, a0, 63
; RV64I-NEXT:
; or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: rori_i64_fshl:
; RV64ZBB-ZBKB:  # %bb.0:
; RV64ZBB-ZBKB-NEXT:  rori a0, a0, 1
; RV64ZBB-ZBKB-NEXT:  ret
%1 = tail call i64 @llvm.fshl.i64(i64 %a, i64 %a, i64 63)
ret i64 %1
}

```

```

define i64 @rori_i64_fshr(i64 %a) nounwind {
; RV64I-LABEL: rori_i64_fshr:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a1, a0, 1

```

```

; RV64I-NEXT:  srli a0, a0, 63
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: rori_i64_fshr:
; RV64ZBB-ZBKB:      # %bb.0:
; RV64ZBB-ZBKB-NEXT:  rori a0, a0, 63
; RV64ZBB-ZBKB-NEXT:  ret
%1 = tail call i64 @llvm.fshr.i64(i64 %a, i64 %a, i64 63)
ret i64 %1
}

```

```

define signext i32 @not_shl_one_i32(i32 signext %x) {
; RV64I-LABEL: not_shl_one_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 1
; RV64I-NEXT:  sllw a0, a1, a0
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: not_shl_one_i32:
; RV64ZBB-ZBKB:      # %bb.0:
; RV64ZBB-ZBKB-NEXT:  li a1,
-2
; RV64ZBB-ZBKB-NEXT:  rolw a0, a1, a0
; RV64ZBB-ZBKB-NEXT:  ret
%1 = shl i32 1, %x
%2 = xor i32 %1, -1
ret i32 %2
}

```

```

define i64 @not_shl_one_i64(i64 %x) {
; RV64I-LABEL: not_shl_one_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 1
; RV64I-NEXT:  sll a0, a1, a0
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: not_shl_one_i64:
; RV64ZBB-ZBKB:      # %bb.0:
; RV64ZBB-ZBKB-NEXT:  li a1, -2
; RV64ZBB-ZBKB-NEXT:  rol a0, a1, a0
; RV64ZBB-ZBKB-NEXT:  ret
%1 = shl i64 1, %x
%2 = xor i64 %1, -1
ret i64 %2
}

```

```

define i8 @srli_i8(i8 %a) nounwind {
; CHECK-LABEL: srli_i8:
; CHECK:      # %bb.0:
; CHECK-NEXT:  slli a0, a0, 56
; CHECK-NEXT:  srli a0, a0, 62
; CHECK-NEXT:  ret
%1 = lshr i8 %a, 6
ret i8 %1
}

```

; We could use sext.b+srai, but slli+srai offers more opportunities for  
; compressed instructions.

```

define i8 @srai_i8(i8 %a) nounwind {
; RV64I-LABEL: srai_i8:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 24
; RV64I-NEXT:
    sraiw a0, a0, 29
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: srai_i8:
; RV64ZBB:    # %bb.0:
; RV64ZBB-NEXT:  slli a0, a0, 56
; RV64ZBB-NEXT:  srai a0, a0, 61
; RV64ZBB-NEXT:  ret
;
; RV64ZBKB-LABEL: srai_i8:
; RV64ZBKB:    # %bb.0:
; RV64ZBKB-NEXT:  slli a0, a0, 24
; RV64ZBKB-NEXT:  sraiw a0, a0, 29
; RV64ZBKB-NEXT:  ret
%1 = ashr i8 %a, 5
ret i8 %1
}

```

; We could use zext.h+srli, but slli+srli offers more opportunities for  
; compressed instructions.

```

define i16 @srli_i16(i16 %a) nounwind {
; CHECK-LABEL: srli_i16:
; CHECK:      # %bb.0:
; CHECK-NEXT:  slli a0, a0, 48
; CHECK-NEXT:  srli a0, a0, 54
; CHECK-NEXT:  ret
%1 = lshr i16 %a, 6
ret i16 %1
}

```



```

; We could use sext.h+srai, but slli+srai offers more opportunities for
; compressed instructions.
define i16 @srai_i16(i16 %a) nounwind {
; RV64I-LABEL: srai_i16:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 16
; RV64I-NEXT:  sraiw a0, a0, 25
; RV64I-NEXT:  ret
;
; RV64ZBB-LABEL: srai_i16:
;
; RV64ZBB:      # %bb.0:
; RV64ZBB-NEXT:  slli a0, a0, 48
; RV64ZBB-NEXT:  srai a0, a0, 57
; RV64ZBB-NEXT:  ret
;
; RV64ZBKB-LABEL: srai_i16:
; RV64ZBKB:      # %bb.0:
; RV64ZBKB-NEXT:  slli a0, a0, 16
; RV64ZBKB-NEXT:  sraiw a0, a0, 25
; RV64ZBKB-NEXT:  ret
%1 = ashr i16 %a, 9
ret i16 %1
}

define i1 @andn_seqz_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: andn_seqz_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  seqz a0, a0
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: andn_seqz_i32:
; RV64ZBB-ZBKB:      # %bb.0:
; RV64ZBB-ZBKB-NEXT:  andn a0, a1, a0
; RV64ZBB-ZBKB-NEXT:  seqz a0, a0
; RV64ZBB-ZBKB-NEXT:  ret
%and = and i32 %a, %b
%cmpeq = icmp eq i32 %and, %b
ret i1 %cmpeq
}

define i1 @andn_seqz_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: andn_seqz_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  xor a0, a0, a1

```

```

; RV64I-NEXT:  seqz a0, a0
; RV64I-NEXT:
    ret
;
; RV64ZBB-ZBKB-LABEL: andn_seqz_i64:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  andn a0, a1, a0
; RV64ZBB-ZBKB-NEXT:  seqz a0, a0
; RV64ZBB-ZBKB-NEXT:  ret
%and = and i64 %a, %b
%cmpeq = icmp eq i64 %and, %b
ret i1 %cmpeq
}

define i1 @andn_snez_i32(i32 signext %a, i32 signext %b) nounwind {
; RV64I-LABEL: andn_snez_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  snez a0, a0
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: andn_snez_i32:
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  andn a0, a1, a0
; RV64ZBB-ZBKB-NEXT:  snez a0, a0
; RV64ZBB-ZBKB-NEXT:  ret
%and = and i32 %a, %b
%cmpeq = icmp ne i32 %and, %b
ret i1 %cmpeq
}

define i1 @andn_snez_i64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: andn_snez_i64:
; RV64I:    # %bb.0:
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  snez a0, a0
; RV64I-NEXT:  ret
;
; RV64ZBB-ZBKB-LABEL: andn_snez_i64:
;
; RV64ZBB-ZBKB:    # %bb.0:
; RV64ZBB-ZBKB-NEXT:  andn a0, a1, a0
; RV64ZBB-ZBKB-NEXT:  snez a0, a0
; RV64ZBB-ZBKB-NEXT:  ret
%and = and i64 %a, %b
%cmpeq = icmp ne i64 %and, %b

```

```

ret i1 %cmpeq
}
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck -check-prefix=RV64I %s
; RUN: llc -mtriple=riscv64 -mattr=+m -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck -check-prefix=RV64IM %s

```

```

define i32 @urem(i32 %a, i32 %b) nounwind {
; RV64I-LABEL: urem:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  srli a1, a1, 32
; RV64I-NEXT:  call __umoddi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: urem:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  remuw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = urem i32 %a, %b
ret i32 %1
}

```

```

define i32 @urem_constant_lhs(i32 %a) nounwind {
;
; RV64I-LABEL: urem_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a1, a0, 32
; RV64I-NEXT:  li a0, 10
; RV64I-NEXT:  call __umoddi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: urem_constant_lhs:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  li a1, 10
; RV64IM-NEXT:  remuw a0, a1, a0

```

```

; RV64IM-NEXT:  ret
%1 = urem i32 10, %a
ret i32 %1
}

define i32 @srem(i32 %a, i32 %b) nounwind {
; RV64I-LABEL: srem:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  sext.w a0, a0
; RV64I-NEXT:  sext.w a1, a1
; RV64I-NEXT:  call __moddi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: srem:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:
    remw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = srem i32 %a, %b
ret i32 %1
}

define i32 @srem_pow2(i32 %a) nounwind {
; RV64I-LABEL: srem_pow2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  sraiw a1, a0, 31
; RV64I-NEXT:  srliw a1, a1, 29
; RV64I-NEXT:  add a1, a0, a1
; RV64I-NEXT:  andi a1, a1, -8
; RV64I-NEXT:  subw a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: srem_pow2:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  sraiw a1, a0, 31
; RV64IM-NEXT:  srliw a1, a1, 29
; RV64IM-NEXT:  add a1, a0, a1
; RV64IM-NEXT:  andi a1, a1, -8
; RV64IM-NEXT:  subw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = srem i32 %a, 8
ret i32 %1
}

```

```

define i32 @srem_pow2_2(i32 %a) nounwind {
; RV64I-LABEL: srem_pow2_2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  sraiw a1, a0, 31
; RV64I-NEXT:  srliw a1, a1, 16
; RV64I-NEXT:  add a1, a0, a1
; RV64I-NEXT:  lui a2, 1048560
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  subw a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: srem_pow2_2:
; RV64IM:
;   # %bb.0:
; RV64IM-NEXT:  sraiw a1, a0, 31
; RV64IM-NEXT:  srliw a1, a1, 16
; RV64IM-NEXT:  add a1, a0, a1
; RV64IM-NEXT:  lui a2, 1048560
; RV64IM-NEXT:  and a1, a1, a2
; RV64IM-NEXT:  subw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = srem i32 %a, 65536
ret i32 %1
}

define i32 @srem_constant_lhs(i32 %a) nounwind {
; RV64I-LABEL: srem_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  sext.w a1, a0
; RV64I-NEXT:  li a0, -10
; RV64I-NEXT:  call __moddi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: srem_constant_lhs:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  li a1, -10
; RV64IM-NEXT:  remw a0, a1, a0
; RV64IM-NEXT:  ret
%1 = srem i32 -10, %a
ret i32 %1
}

define i64 @urem64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: urem64:

```

```

; RV64I:    # %bb.0:
; RV64I-NEXT:  tail __umoddi3
;
; RV64IM-LABEL:
urem64:
; RV64IM:    # %bb.0:
; RV64IM-NEXT:  remu a0, a0, a1
; RV64IM-NEXT:  ret
%1 = urem i64 %a, %b
ret i64 %1
}

define i64 @urem64_constant_lhs(i64 %a) nounwind {
; RV64I-LABEL: urem64_constant_lhs:
; RV64I:    # %bb.0:
; RV64I-NEXT:  mv a1, a0
; RV64I-NEXT:  li a0, 10
; RV64I-NEXT:  tail __umoddi3
;
; RV64IM-LABEL: urem64_constant_lhs:
; RV64IM:    # %bb.0:
; RV64IM-NEXT:  li a1, 10
; RV64IM-NEXT:  remu a0, a1, a0
; RV64IM-NEXT:  ret
%1 = urem i64 10, %a
ret i64 %1
}

define i64 @srem64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: srem64:
; RV64I:    # %bb.0:
; RV64I-NEXT:  tail __moddi3
;
; RV64IM-LABEL: srem64:
; RV64IM:    # %bb.0:
; RV64IM-NEXT:  rem a0, a0, a1
; RV64IM-NEXT:  ret
%1 = srem i64 %a, %b
ret i64 %1
}

define i64 @srem64_constant_lhs(i64 %a) nounwind {
; RV64I-LABEL: srem64_constant_lhs:
; RV64I:    # %bb.0:
; RV64I-NEXT:  mv a1, a0
; RV64I-NEXT:  li a0, -10
; RV64I-NEXT:  tail __moddi3
;

```

```

; RV64IM-LABEL: srem64_constant_lhs:
;
RV64IM:    # %bb.0:
; RV64IM-NEXT:  li a1, -10
; RV64IM-NEXT:  rem a0, a1, a0
; RV64IM-NEXT:  ret
%1 = srem i64 -10, %a
ret i64 %1
}

define i8 @urem8(i8 %a, i8 %b) nounwind {
; RV64I-LABEL: urem8:
; RV64I:    # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  andi a0, a0, 255
; RV64I-NEXT:  andi a1, a1, 255
; RV64I-NEXT:  call __umoddi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: urem8:
; RV64IM:    # %bb.0:
; RV64IM-NEXT:  andi a1, a1, 255
; RV64IM-NEXT:  andi a0, a0, 255
; RV64IM-NEXT:  remuw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = urem i8 %a, %b
ret i8 %1
}

define i8 @urem8_constant_lhs(i8 %a) nounwind {
; RV64I-LABEL: urem8_constant_lhs:
; RV64I:    # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  andi a1, a0, 255
; RV64I-NEXT:
    li a0, 10
; RV64I-NEXT:  call __umoddi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: urem8_constant_lhs:
; RV64IM:    # %bb.0:
; RV64IM-NEXT:  andi a0, a0, 255

```

```

; RV64IM-NEXT:  li a1, 10
; RV64IM-NEXT:  remuw a0, a1, a0
; RV64IM-NEXT:  ret
%1 = urem i8 10, %a
ret i8 %1
}

```

```

define i8 @srem8(i8 %a, i8 %b) nounwind {
; RV64I-LABEL: srem8:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a1, a1, 24
; RV64I-NEXT:  sraiw a1, a1, 24
; RV64I-NEXT:  slli a0, a0, 24
; RV64I-NEXT:  sraiw a0, a0, 24
; RV64I-NEXT:  call __moddi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: srem8:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  slli a1, a1, 24
; RV64IM-NEXT:  sraiw a1, a1, 24
; RV64IM-NEXT:  slli a0, a0, 24
; RV64IM-NEXT:  sraiw
a0, a0, 24
; RV64IM-NEXT:  remw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = srem i8 %a, %b
ret i8 %1
}

```

```

define i8 @srem8_constant_lhs(i8 %a) nounwind {
; RV64I-LABEL: srem8_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a0, a0, 24
; RV64I-NEXT:  sraiw a1, a0, 24
; RV64I-NEXT:  li a0, -10
; RV64I-NEXT:  call __moddi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;

```



```

; RV64IM-LABEL: srem8_constant_lhs:
; RV64IM:      # %bb.0:
; RV64IM-NEXT: slli a0, a0, 24
; RV64IM-NEXT: sraiw a0, a0, 24
; RV64IM-NEXT: li a1, -10
; RV64IM-NEXT: remw a0, a1, a0
; RV64IM-NEXT: ret
%1 = srem i8 -10, %a
ret i8 %1
}

```

```

define i16 @urem16(i16 %a, i16 %b) nounwind {
; RV64I-LABEL: urem16:
; RV64I:      # %bb.0:
; RV64I-NEXT: addi sp, sp, -16
; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT: lui a2, 16
; RV64I-NEXT:
    addiw a2, a2, -1
; RV64I-NEXT: and a0, a0, a2
; RV64I-NEXT: and a1, a1, a2
; RV64I-NEXT: call __umoddi3
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT: addi sp, sp, 16
; RV64I-NEXT: ret
;
; RV64IM-LABEL: urem16:
; RV64IM:      # %bb.0:
; RV64IM-NEXT: lui a2, 16
; RV64IM-NEXT: addi a2, a2, -1
; RV64IM-NEXT: and a1, a1, a2
; RV64IM-NEXT: and a0, a0, a2
; RV64IM-NEXT: remuw a0, a0, a1
; RV64IM-NEXT: ret
%1 = urem i16 %a, %b
ret i16 %1
}

```

```

define i16 @urem16_constant_lhs(i16 %a) nounwind {
; RV64I-LABEL: urem16_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT: addi sp, sp, -16
; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT: slli a0, a0, 48
; RV64I-NEXT: srli a1, a0, 48
; RV64I-NEXT: li a0, 10
; RV64I-NEXT: call __umoddi3

```

```

; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT: addi sp, sp, 16
; RV64I-NEXT: ret
;
; RV64IM-LABEL: urem16_constant_lhs:
; RV64IM:    # %bb.0:
;
RV64IM-NEXT: slli a0, a0, 48
; RV64IM-NEXT: srli a0, a0, 48
; RV64IM-NEXT: li a1, 10
; RV64IM-NEXT: remuw a0, a1, a0
; RV64IM-NEXT: ret
%1 = urem i16 10, %a
ret i16 %1
}

```

```

define i16 @srem16(i16 %a, i16 %b) nounwind {
; RV64I-LABEL: srem16:
; RV64I:    # %bb.0:
; RV64I-NEXT: addi sp, sp, -16
; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT: slli a1, a1, 16
; RV64I-NEXT: sraiw a1, a1, 16
; RV64I-NEXT: slli a0, a0, 16
; RV64I-NEXT: sraiw a0, a0, 16
; RV64I-NEXT: call __moddi3
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT: addi sp, sp, 16
; RV64I-NEXT: ret
;
; RV64IM-LABEL: srem16:
; RV64IM:    # %bb.0:
; RV64IM-NEXT: slli a1, a1, 16
; RV64IM-NEXT: sraiw a1, a1, 16
; RV64IM-NEXT: slli a0, a0, 16
; RV64IM-NEXT: sraiw a0, a0, 16
; RV64IM-NEXT: remw a0, a0, a1
; RV64IM-NEXT: ret
%1 = srem i16 %a, %b
ret i16 %1
}

```

```

define i16 @srem16_constant_lhs(i16 %a) nounwind {
; RV64I-LABEL: srem16_constant_lhs:
;
RV64I:    # %bb.0:
; RV64I-NEXT: addi sp, sp, -16
; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill

```

```

; RV64I-NEXT:  slli a0, a0, 16
; RV64I-NEXT:  sraiw a1, a0, 16
; RV64I-NEXT:  li a0, -10
; RV64I-NEXT:  call __moddi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: srem16_constant_lhs:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  slli a0, a0, 16
; RV64IM-NEXT:  sraiw a0, a0, 16
; RV64IM-NEXT:  li a1, -10
; RV64IM-NEXT:  remw a0, a1, a0
; RV64IM-NEXT:  ret
%1 = srem i16 -10, %a
ret i16 %1
}
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=aarch64-apple-ios %s -o - | FileCheck %s

```

```

define <16 x double> @test_sitofp_fixed(<16 x i32> %in) {

```

```

; CHECK-LABEL: test_sitofp_fixed:
; CHECK:      ; %bb.0:
; CHECK-NEXT:  sshll2.2d v4, v0, #0
; CHECK-NEXT:  sshll.2d v0, v0, #0
; CHECK-NEXT:  sshll2.2d v5, v1, #0
; CHECK-NEXT:  sshll.2d v6, v1, #0
; CHECK-NEXT:  sshll.2d v7, v2, #0
; CHECK-NEXT:  sshll2.2d v16, v2, #0
; CHECK-NEXT:  sshll2.2d v17, v3, #0
; CHECK-NEXT:  sshll.2d v18, v3, #0
; CHECK-NEXT:  scvtf.2d v1, v4, #6
; CHECK-NEXT:  scvtf.2d v0, v0, #6
; CHECK-NEXT:  scvtf.2d v3, v5, #6
; CHECK-NEXT:  scvtf.2d v2, v6, #6
; CHECK-NEXT:  scvtf.2d v4, v7, #6
; CHECK-NEXT:  scvtf.2d v5, v16, #6
; CHECK-NEXT:  scvtf.2d v7, v17, #6
; CHECK-NEXT:  scvtf.2d v6, v18, #6
; CHECK-NEXT:  ret

```

```

%flt = sitofp <16 x i32> %in to <16 x double>
%res = fdiv <16 x double> %flt, <double 64.0, double 64.0, double 64.0,
double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double
64.0, double 64.0, double 64.0, double 64.0, double 64.0>
ret <16 x double> %res
}

```

```

; This one is small enough to satisfy isSimple, but still illegally large.
define <4 x double> @test_sitofp_fixed_shortish(<4 x i64> %in) {
; CHECK-LABEL: test_sitofp_fixed_shortish:
; CHECK:      ; %bb.0:
; CHECK-NEXT:  scvtf.2d v0, v0, #6
; CHECK-NEXT:  scvtf.2d v1, v1, #6
; CHECK-NEXT:  ret

%flt = sitofp <4 x i64> %in to <4 x double>
%res = fdiv <4 x double> %flt, <double 64.0, double 64.0, double 64.0, double 64.0>
ret <4 x double> %res
}
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64I

```

```

; These tests are each targeted at a particular RISC-V ALU instruction. Most
; other files in this folder exercise LLVM IR instructions that don't directly
; match a RISC-V instruction.

```

```

; Register-immediate instructions.

```

```

define i32 @addi(i32 %a) nounwind {
; RV64I-LABEL: addi:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addiw a0, a0, 1
; RV64I-NEXT:  ret
%1 = add i32 %a, 1
ret i32 %1
}

```

```

define i32 @slti(i32 %a) nounwind {
; RV64I-LABEL: slti:
; RV64I:      # %bb.0:
; RV64I-NEXT:  sext.w a0, a0
; RV64I-NEXT:  slti a0, a0, 2
; RV64I-NEXT:  ret
%1 = icmp slt i32 %a, 2
%2 = zext i1 %1 to i32
ret i32 %2
}

```

```

define i32 @sltiu(i32 %a) nounwind {
; RV64I-LABEL: sltiu:
; RV64I:      # %bb.0:
; RV64I-NEXT:  sext.w a0, a0

```

```

; RV64I-NEXT:  sltiu a0, a0, 3
; RV64I-NEXT:
    ret
%1 = icmp ult i32 %a, 3
%2 = zext i1 %1 to i32
ret i32 %2
}

define i32 @xori(i32 %a) nounwind {
; RV64I-LABEL: xori:
; RV64I:      # %bb.0:
; RV64I-NEXT:  xori a0, a0, 4
; RV64I-NEXT:  ret
%1 = xor i32 %a, 4
ret i32 %1
}

define i32 @ori(i32 %a) nounwind {
; RV64I-LABEL: ori:
; RV64I:      # %bb.0:
; RV64I-NEXT:  ori a0, a0, 5
; RV64I-NEXT:  ret
%1 = or i32 %a, 5
ret i32 %1
}

define i32 @andi(i32 %a) nounwind {
; RV64I-LABEL: andi:
; RV64I:      # %bb.0:
; RV64I-NEXT:  andi a0, a0, 6
; RV64I-NEXT:  ret
%1 = and i32 %a, 6
ret i32 %1
}

define i32 @slli(i32 %a) nounwind {
; RV64I-LABEL: slli:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slliw a0, a0, 7
; RV64I-NEXT:  ret
%1 = shl i32 %a, 7
ret i32 %1
}

define i32 @srli(i32 %a) nounwind {
; RV64I-LABEL: srli:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srliw a0, a0, 8

```

```

; RV64I-NEXT:  ret
%1 = lshr i32 %a, 8
ret i32 %1
}

define i32 @srai(i32 %a) nounwind {
; RV64I-LABEL: srai:
; RV64I:
# %bb.0:
; RV64I-NEXT:  sraiw a0, a0, 9
; RV64I-NEXT:  ret
%1 = ashr i32 %a, 9
ret i32 %1
}

; Register-register instructions

define i32 @add(i32 %a, i32 %b) nounwind {
; RV64I-LABEL: add:
; RV64I:  # %bb.0:
; RV64I-NEXT:  addw a0, a0, a1
; RV64I-NEXT:  ret
%1 = add i32 %a, %b
ret i32 %1
}

define i32 @sub(i32 %a, i32 %b) nounwind {
; RV64I-LABEL: sub:
; RV64I:  # %bb.0:
; RV64I-NEXT:  subw a0, a0, a1
; RV64I-NEXT:  ret
%1 = sub i32 %a, %b
ret i32 %1
}

define i32 @sub_negative_constant_lhs(i32 %a) nounwind {
; RV64I-LABEL: sub_negative_constant_lhs:
; RV64I:  # %bb.0:
; RV64I-NEXT:  li a1, -2
; RV64I-NEXT:  subw a0, a1, a0
; RV64I-NEXT:  ret
%1 = sub i32 -2, %a
ret i32 %1
}

define i32 @sll(i32 %a, i32 %b) nounwind {
; RV64I-LABEL: sll:
; RV64I:  # %bb.0:

```

```
; RV64I-NEXT:  sllw a0, a0, a1
; RV64I-NEXT:  ret
%1 = shl i32 %a, %b
ret i32 %1
}
```

; Make sure we don't emit instructions to zero extend the shift amount to i64.

```
define
i32 @sll_shamt_zext(i32 %a, i32 %b) nounwind {
; RV64I-LABEL: sll_shamt_zext:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi a1, a1, 1
; RV64I-NEXT:  sllw a0, a0, a1
; RV64I-NEXT:  ret
%shamt = add i32 %b, 1
%1 = shl i32 %a, %shamt
ret i32 %1
}
```

```
define i32 @sll_negative_constant_lhs(i32 %a) nounwind {
; RV64I-LABEL: sll_negative_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, -1
; RV64I-NEXT:  sllw a0, a1, a0
; RV64I-NEXT:  ret
%1 = shl i32 -1, %a
ret i32 %1
}
```

```
define i32 @slt(i32 %a, i32 %b) nounwind {
; RV64I-LABEL: slt:
; RV64I:      # %bb.0:
; RV64I-NEXT:  sext.w a1, a1
; RV64I-NEXT:  sext.w a0, a0
; RV64I-NEXT:  slt a0, a0, a1
; RV64I-NEXT:  ret
%1 = icmp slt i32 %a, %b
%2 = zext i1 %1 to i32
ret i32 %2
}
```

```
define i32 @sltu(i32 %a, i32 %b) nounwind {
;
; RV64I-LABEL: sltu:
; RV64I:      # %bb.0:
; RV64I-NEXT:  sext.w a1, a1
; RV64I-NEXT:  sext.w a0, a0
; RV64I-NEXT:  sltu a0, a0, a1
```

```
; RV64I-NEXT:  ret
%1 = icmp ult i32 %a,
%b
%2 = zext i1 %1 to i32
ret i32 %2
}
```

```
define i32 @xor(i32 %a, i32 %b) nounwind {
;
; RV64I-LABEL: xor:
; RV64I:      # %bb.0:
; RV64I-NEXT:  xor a0, a0, a1
; RV64I-NEXT:  ret
%1 = xor i32 %a, %b
ret i32 %1
}
```

```
define i32 @srl(i32 %a, i32 %b) nounwind {
;
; RV64I-LABEL: srl:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srlw a0, a0, a1
; RV64I-NEXT:  ret
%1 = lshr i32 %a, %b
ret i32 %1
}
```

; Make sure we don't emit instructions to zero extend the shift amount to i64.

```
define i32 @srl_shamt_zext(i32 %a, i32 %b) nounwind {
; RV64I-LABEL: srl_shamt_zext:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi a1, a1, 1
; RV64I-NEXT:  srlw a0, a0, a1
; RV64I-NEXT:  ret
%shamt = add i32 %b, 1
%1 = lshr i32 %a, %shamt
ret i32 %1
}
```

```
define i32 @srl_negative_constant_lhs(i32 %a) nounwind {
;
; RV64I-LABEL: srl_negative_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, -1
; RV64I-NEXT:  srlw a0, a1, a0
; RV64I-NEXT:  ret
%1 = lshr i32 -1, %a
ret i32 %1
}
```



```
}
```

```
define
```

```
  i32 @sra(i32 %a, i32 %b) nounwind {
```

```
  ;
```

```
  ; RV64I-LABEL: sra:
```

```
  ; RV64I:      # %bb.0:
```

```
  ; RV64I-NEXT:  sraw a0, a0, a1
```

```
  ; RV64I-NEXT:  ret
```

```
    %1 = ashr i32 %a, %b
```

```
    ret i32 %1
```

```
}
```

```
; Make sure we don't emit instructions to zero extend the shift amount to i64.
```

```
define i32 @sra_shamt_zext(i32 %a, i32 %b) nounwind {
```

```
  ; RV64I-LABEL: sra_shamt_zext:
```

```
  ; RV64I:      # %bb.0:
```

```
  ; RV64I-NEXT:  addi a1, a1, 1
```

```
  ; RV64I-NEXT:  sraw a0, a0, a1
```

```
  ; RV64I-NEXT:  ret
```

```
    %shamt = add i32 %b, 1
```

```
    %1 = ashr i32 %a, %shamt
```

```
    ret i32 %1
```

```
}
```

```
define i32 @sra_negative_constant_lhs(i32 %a) nounwind {
```

```
  ;
```

```
  ; RV64I-LABEL: sra_negative_constant_lhs:
```

```
  ; RV64I:      # %bb.0:
```

```
  ; RV64I-NEXT:  lui a1, 524288
```

```
  ; RV64I-NEXT:  sraw a0, a1, a0
```

```
  ; RV64I-NEXT:  ret
```

```
    %1 = ashr i32 2147483648, %a
```

```
    ret i32 %1
```

```
}
```

```
define i32 @or(i32 %a, i32 %b) nounwind {
```

```
  ;
```

```
  ; RV64I-LABEL: or:
```

```
  ; RV64I:      # %bb.0:
```

```
  ; RV64I-NEXT:  or a0, a0, a1
```

```
  ; RV64I-NEXT:  ret
```

```
    %1 = or i32 %a, %b
```

```
    ret i32 %1
```

```
}
```

```
define i32 @and(i32 %a, i32 %b) nounwind {
```

```
  ;
```

```

; RV64I-LABEL:
and:
; RV64I:    # %bb.0:
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  ret
%1 = and i32 %a, %b
ret i32 %1
}
; NOTE: Assertions have been autogenerated by utils/update_test_checks.py UTC_ARGS: --include-generated-funcs
; RUN: opt -S -passes=verify,iroutliner -ir-outlining-no-cost < %s | FileCheck %s

```

; This test checks that we do outline indirect calls when it is not specified  
; that we should not.

```

declare void @f1(ptr, ptr);
declare void @f2(ptr, ptr);

```

```

define void @function1(ptr %func) {
entry:
%a = alloca i32, align 4
%b = alloca i32, align 4
%c = alloca i32, align 4
store i32 2, ptr %a, align 4
store i32 3, ptr %b, align 4
store i32 4, ptr %c, align 4
call void %func()
%al = load i32, ptr %a
%bl = load i32, ptr %b
%cl = load i32, ptr %c
ret void
}

```

```

define void @function2(ptr %func) {
entry:
%a = alloca i32, align 4
%b = alloca i32, align 4
%c = alloca i32, align 4
store i32 2, ptr %a, align 4
store i32 3, ptr %b, align 4
store i32 4, ptr %c, align 4
call void %func()
%al = load i32, ptr %a
%bl = load i32, ptr %b
%cl = load i32, ptr %c
ret void
}
;

```

```

CHECK-LABEL: @function1(
; CHECK-NEXT: entry:
; CHECK-NEXT: [[A:%.*]] = alloca i32, align 4
; CHECK-NEXT: [[B:%.*]] = alloca i32, align 4
; CHECK-NEXT: [[C:%.*]] = alloca i32, align 4
; CHECK-NEXT: call void @outlined_ir_func_0(ptr [[A]], ptr [[B]], ptr [[C]], ptr [[FUNC:%.*]])
; CHECK-NEXT: ret void
;
;
; CHECK-LABEL: @function2(
; CHECK-NEXT: entry:
; CHECK-NEXT: [[A:%.*]] = alloca i32, align 4
; CHECK-NEXT: [[B:%.*]] = alloca i32, align 4
; CHECK-NEXT: [[C:%.*]] = alloca i32, align 4
; CHECK-NEXT: call void @outlined_ir_func_0(ptr [[A]], ptr [[B]], ptr [[C]], ptr [[FUNC:%.*]])
; CHECK-NEXT: ret void
;
;
; CHECK-LABEL: define internal void @outlined_ir_func_0(
; CHECK-NEXT: newFuncRoot:
; CHECK-NEXT: br label [[ENTRY_TO_OUTLINE:%.*]]
; CHECK: entry_to_outline:
; CHECK-NEXT: store i32 2, ptr [[TMP0:%.*]], align 4
; CHECK-NEXT: store i32 3, ptr [[TMP1:%.*]], align 4
; CHECK-NEXT: store i32 4, ptr [[TMP2:%.*]], align 4
; CHECK-NEXT:
; CHECK-NEXT: call void [[TMP3:%.*]]()
; CHECK-NEXT: [[AL:%.*]] = load i32, ptr [[TMP0]], align 4
; CHECK-NEXT: [[BL:%.*]] = load i32, ptr [[TMP1]], align 4
; CHECK-NEXT: [[CL:%.*]] = load i32, ptr [[TMP2]], align 4
; CHECK-NEXT: br label [[ENTRY_AFTER_OUTLINE_EXITSTUB:%.*]]
; CHECK: entry_after_outline.exitStub:
; CHECK-NEXT: ret void
;
;
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64I
; RUN: llc -mtriple=riscv64 -mattr=+xtheadbb -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64XTHEADBB

```

```

declare i32 @llvm.ctlz.i32(i32, i1)

```

```

define signext i32 @ctlz_i32(i32 signext %a) nounwind {
; RV64I-LABEL: ctlz_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT: beqz a0, .LBB0_2
; RV64I-NEXT: # %bb.1: # %cond.false
; RV64I-NEXT: addi sp, sp, -16

```

```

; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT: srliw a1, a0, 1
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 2
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 4
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 8
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 16
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT:
    not a0, a0
; RV64I-NEXT: srliw a1, a0, 1
; RV64I-NEXT: lui a2, 349525
; RV64I-NEXT: addi a2, a2, 1365
; RV64I-NEXT: and a1, a1, a2
; RV64I-NEXT: subw a0, a0, a1
; RV64I-NEXT: lui a1, 209715
; RV64I-NEXT: addi a1, a1, 819
; RV64I-NEXT: and a2, a0, a1
; RV64I-NEXT: srliw a0, a0, 2
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: add a0, a2, a0
; RV64I-NEXT: srliw a1, a0, 4
; RV64I-NEXT: addw a0, a0, a1
; RV64I-NEXT: lui a1, 61681
; RV64I-NEXT: addiw a1, a1, -241
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: lui a1, 4112
; RV64I-NEXT: addiw a1, a1, 257
; RV64I-NEXT: call __muldi3
; RV64I-NEXT: srliw a0, a0, 24
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT: addi sp, sp, 16
; RV64I-NEXT: ret
; RV64I-NEXT: .LBB0_2:
; RV64I-NEXT: li a0, 32
; RV64I-NEXT: ret
;
; RV64XTHEADBB-LABEL: ctlz_i32:
; RV64XTHEADBB:    # %bb.0:
; RV64XTHEADBB-NEXT: th.extu a0, a0, 31, 0
; RV64XTHEADBB-NEXT: th.ff1 a0,
a0
; RV64XTHEADBB-NEXT: addi a0, a0, -32
; RV64XTHEADBB-NEXT: ret
%1 = call i32 @llvm.ctlz.i32(i32 %a, i1 false)

```

```

ret i32 %1
}

define signext i32 @log2_i32(i32 signext %a) nounwind {
; RV64I-LABEL: log2_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT: beqz a0, .LBB1_2
; RV64I-NEXT: # %bb.1: # %cond.false
; RV64I-NEXT: addi sp, sp, -16
; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT: srliw a1, a0, 1
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 2
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 4
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 8
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 16
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: not a0, a0
; RV64I-NEXT: srliw a1, a0, 1
; RV64I-NEXT: lui a2, 349525
; RV64I-NEXT: addi a2, a2, 1365
; RV64I-NEXT: and a1, a1, a2
; RV64I-NEXT: subw a0, a0, a1
; RV64I-NEXT: lui a1, 209715
; RV64I-NEXT: addi a1, a1, 819
; RV64I-NEXT:
and a2, a0, a1
; RV64I-NEXT: srliw a0, a0, 2
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: add a0, a2, a0
; RV64I-NEXT: srliw a1, a0, 4
; RV64I-NEXT: addw a0, a0, a1
; RV64I-NEXT: lui a1, 61681
; RV64I-NEXT: addiw a1, a1, -241
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: lui a1, 4112
; RV64I-NEXT: addiw a1, a1, 257
; RV64I-NEXT: call __muldi3
; RV64I-NEXT: srliw a0, a0, 24
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT: addi sp, sp, 16
; RV64I-NEXT: j .LBB1_3
; RV64I-NEXT: .LBB1_2:
; RV64I-NEXT: li a0, 32
; RV64I-NEXT: .LBB1_3: # %cond.end

```

```

; RV64I-NEXT:  li a1, 31
; RV64I-NEXT:  subw a0, a1, a0
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: log2_i32:
; RV64XTHEADBB:    # %bb.0:
; RV64XTHEADBB-NEXT:  th.extu a0, a0, 31, 0
; RV64XTHEADBB-NEXT:  th.ff1 a0, a0
; RV64XTHEADBB-NEXT:  addi a0, a0, -32
; RV64XTHEADBB-NEXT:  li a1, 31
; RV64XTHEADBB-NEXT:  subw a0, a1, a0
; RV64XTHEADBB-NEXT:  ret
%1 = call i32 @llvmctlz.i32(i32
%a, i1 false)
%2 = sub i32 31, %1
ret i32 %2
}

define signext i32 @log2_ceil_i32(i32 signext %a) nounwind {
; RV64I-LABEL: log2_ceil_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  sd s0, 0(sp) # 8-byte Folded Spill
; RV64I-NEXT:  addiw a0, a0, -1
; RV64I-NEXT:  li s0, 32
; RV64I-NEXT:  li a1, 32
; RV64I-NEXT:  beqz a0, .LBB2_2
; RV64I-NEXT:  # %bb.1: # %cond.false
; RV64I-NEXT:  srliw a1, a0, 1
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 2
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 4
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 8
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srliw a1, a0, 16
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  srliw a1, a0, 1
; RV64I-NEXT:  lui a2, 349525
; RV64I-NEXT:  addi a2, a2, 1365
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  subw a0, a0, a1
; RV64I-NEXT:
    lui a1, 209715
; RV64I-NEXT:  addi a1, a1, 819

```

```

; RV64I-NEXT: and a2, a0, a1
; RV64I-NEXT: srliw a0, a0, 2
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: add a0, a2, a0
; RV64I-NEXT: srliw a1, a0, 4
; RV64I-NEXT: addw a0, a0, a1
; RV64I-NEXT: lui a1, 61681
; RV64I-NEXT: addiw a1, a1, -241
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: lui a1, 4112
; RV64I-NEXT: addiw a1, a1, 257
; RV64I-NEXT: call __muldi3
; RV64I-NEXT: srliw a1, a0, 24
; RV64I-NEXT: .LBB2_2: # %cond.end
; RV64I-NEXT: subw a0, s0, a1
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT: ld s0, 0(sp) # 8-byte Folded Reload
; RV64I-NEXT: addi sp, sp, 16
; RV64I-NEXT: ret
;
; RV64XTHEADBB-LABEL: log2_ceil_i32:
; RV64XTHEADBB:    # %bb.0:
; RV64XTHEADBB-NEXT: addi a0, a0, -1
; RV64XTHEADBB-NEXT: slli a0, a0, 32
; RV64XTHEADBB-NEXT: srli a0, a0, 32
; RV64XTHEADBB-NEXT: th.ff1 a0, a0
; RV64XTHEADBB-NEXT: addi a0, a0, -32
; RV64XTHEADBB-NEXT:
    li a1, 32
; RV64XTHEADBB-NEXT: subw a0, a1, a0
; RV64XTHEADBB-NEXT: ret
%1 = sub i32 %a, 1
%2 = call i32 @llvmctlz.i32(i32 %1, i1 false)
%3 = sub i32 32, %2
ret i32 %3
}

define signext i32 @findLastSet_i32(i32 signext %a) nounwind {
; RV64I-LABEL: findLastSet_i32:
; RV64I:    # %bb.0:
; RV64I-NEXT: addi sp, sp, -16
; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT: sd s0, 0(sp) # 8-byte Folded Spill
; RV64I-NEXT: mv s0, a0
; RV64I-NEXT: srliw a0, a0, 1
; RV64I-NEXT: or a0, s0, a0
; RV64I-NEXT: srliw a1, a0, 2
; RV64I-NEXT: or a0, a0, a1

```

```

; RV64I-NEXT: srliw a1, a0, 4
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 8
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: srliw a1, a0, 16
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: not a0, a0
; RV64I-NEXT: srliw a1, a0, 1
; RV64I-NEXT: lui a2, 349525
; RV64I-NEXT: addi a2, a2, 1365
; RV64I-NEXT: and a1, a1, a2
; RV64I-NEXT: subw a0, a0, a1
; RV64I-NEXT:
    lui a1, 209715
; RV64I-NEXT: addi a1, a1, 819
; RV64I-NEXT: and a2, a0, a1
; RV64I-NEXT: srliw a0, a0, 2
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: add a0, a2, a0
; RV64I-NEXT: srliw a1, a0, 4
; RV64I-NEXT: addw a0, a0, a1
; RV64I-NEXT: lui a1, 61681
; RV64I-NEXT: addiw a1, a1, -241
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: lui a1, 4112
; RV64I-NEXT: addiw a1, a1, 257
; RV64I-NEXT: call __muldi3
; RV64I-NEXT: srliw a0, a0, 24
; RV64I-NEXT: xori a0, a0, 31
; RV64I-NEXT: snez a1, s0
; RV64I-NEXT: addiw a1, a1, -1
; RV64I-NEXT: or a0, a1, a0
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT: ld s0, 0(sp) # 8-byte Folded Reload
; RV64I-NEXT: addi sp, sp, 16
; RV64I-NEXT: ret
;
; RV64XTHEADBB-LABEL: findLastSet_i32:
; RV64XTHEADBB:    # %bb.0:
; RV64XTHEADBB-NEXT: th.extu a1, a0, 31, 0
; RV64XTHEADBB-NEXT: th.ff1 a1, a1
; RV64XTHEADBB-NEXT: addiw a1, a1, -32
; RV64XTHEADBB-NEXT: xori a1, a1,
31
; RV64XTHEADBB-NEXT: snez a0, a0
; RV64XTHEADBB-NEXT: addiw a0, a0, -1
; RV64XTHEADBB-NEXT: or a0, a0, a1
; RV64XTHEADBB-NEXT: ret

```



```

%1 = call i32 @llvm.ctlz.i32(i32 %a, i1 true)
%2 = xor i32 31, %1
%3 = icmp eq i32 %a, 0
%4 = select i1 %3, i32 -1, i32 %2
ret i32 %4
}

```

```

define i32 @ctlz_lshr_i32(i32 signext %a) {
; RV64I-LABEL: ctlz_lshr_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srlw a0, a0, 1
; RV64I-NEXT:  beqz a0, .LBB4_2
; RV64I-NEXT:  # %bb.1: # %cond.false
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  .cfi_def_cfa_offset 16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  .cfi_offset ra, -8
; RV64I-NEXT:  srlw a1, a0, 1
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srlw a1, a0, 2
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srlw a1, a0, 4
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srlw a1, a0, 8
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srlw a1, a0, 16
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  not
a0, a0
; RV64I-NEXT:  srlw a1, a0, 1
; RV64I-NEXT:  lui a2, 349525
; RV64I-NEXT:  addi a2, a2, 1365
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  subw a0, a0, a1
; RV64I-NEXT:  lui a1, 209715
; RV64I-NEXT:  addi a1, a1, 819
; RV64I-NEXT:  and a2, a0, a1
; RV64I-NEXT:  srlw a0, a0, 2
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  add a0, a2, a0
; RV64I-NEXT:  srlw a1, a0, 4
; RV64I-NEXT:  addw a0, a0, a1
; RV64I-NEXT:  lui a1, 61681
; RV64I-NEXT:  addiw a1, a1, -241
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  lui a1, 4112
; RV64I-NEXT:  addiw a1, a1, 257
; RV64I-NEXT:  call __muldi3

```

```

; RV64I-NEXT:  srlw a0, a0, 24
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
; RV64I-NEXT:  .LBB4_2:
; RV64I-NEXT:  li a0, 32
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: ctlz_lshr_i32:
; RV64XTHEADBB:    # %bb.0:
; RV64XTHEADBB-NEXT:  srlw a0, a0, 1
; RV64XTHEADBB-NEXT:  th.ff1 a0, a0
; RV64XTHEADBB-NEXT:
    addi a0, a0, -32
; RV64XTHEADBB-NEXT:  ret
%1 = lshr i32 %a, 1
%2 = call i32 @llvm.ctlz.i32(i32 %1, i1 false)
ret i32 %2
}

```

```
declare i64 @llvm.ctlz.i64(i64, i1)
```

```

define i64 @ctlz_i64(i64 %a) nounwind {
; RV64I-LABEL: ctlz_i64:
; RV64I:    # %bb.0:
; RV64I-NEXT:  beqz a0, .LBB5_2
; RV64I-NEXT:  # %bb.1: # %cond.false
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  srli a1, a0, 1
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srli a1, a0, 2
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srli a1, a0, 4
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srli a1, a0, 8
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srli a1, a0, 16
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  srli a1, a0, 32
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  not a0, a0
; RV64I-NEXT:  srli a1, a0, 1
; RV64I-NEXT:  lui a2, 349525
; RV64I-NEXT:  addiw a2, a2, 1365
; RV64I-NEXT:  slli a3, a2, 32
; RV64I-NEXT:  add a2, a2,

```

```

a3
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  sub a0, a0, a1
; RV64I-NEXT:  lui a1, 209715
; RV64I-NEXT:  addiw a1, a1, 819
; RV64I-NEXT:  slli a2, a1, 32
; RV64I-NEXT:  add a1, a1, a2
; RV64I-NEXT:  and a2, a0, a1
; RV64I-NEXT:  srli a0, a0, 2
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  add a0, a2, a0
; RV64I-NEXT:  srli a1, a0, 4
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  lui a1, 61681
; RV64I-NEXT:  addiw a1, a1, -241
; RV64I-NEXT:  slli a2, a1, 32
; RV64I-NEXT:  add a1, a1, a2
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  lui a1, 4112
; RV64I-NEXT:  addiw a1, a1, 257
; RV64I-NEXT:  slli a2, a1, 32
; RV64I-NEXT:  add a1, a1, a2
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:  srli a0, a0, 56
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
; RV64I-NEXT:  .LBB5_2:
; RV64I-NEXT:  li a0, 64
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: ctlz_i64:
; RV64XTHEADBB:      # %bb.0:
; RV64XTHEADBB-NEXT:
    th.ff1 a0, a0
; RV64XTHEADBB-NEXT:  ret
%1 = call i64 @llvm.ctlz.i64(i64 %a, i1 false)
ret i64 %1
}

```

```

declare i32 @llvm.cttz.i32(i32, i1)

```

```

define signext i32 @cttz_i32(i32 signext %a) nounwind {
; RV64I-LABEL: cttz_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  beqz a0, .LBB6_2
; RV64I-NEXT:  # %bb.1: # %cond.false
; RV64I-NEXT:  addi sp, sp, -16

```

```

; RV64I-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT: negw a1, a0
; RV64I-NEXT: and a0, a0, a1
; RV64I-NEXT: lui a1, 30667
; RV64I-NEXT: addiw a1, a1, 1329
; RV64I-NEXT: call __muldi3
; RV64I-NEXT: srliw a0, a0, 27
; RV64I-NEXT: lui a1, %hi(.LCPI6_0)
; RV64I-NEXT: addi a1, a1, %lo(.LCPI6_0)
; RV64I-NEXT: add a0, a1, a0
; RV64I-NEXT: lbu a0, 0(a0)
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT: addi sp, sp, 16
; RV64I-NEXT: ret
; RV64I-NEXT: .LBB6_2:
; RV64I-NEXT: li a0, 32
; RV64I-NEXT: ret
;
; RV64XTHEADBB-LABEL: cttz_i32:
; RV64XTHEADBB: #
%bb.0:
; RV64XTHEADBB-NEXT: beqz a0, .LBB6_2
; RV64XTHEADBB-NEXT: # %bb.1: # %cond.false
; RV64XTHEADBB-NEXT: addi sp, sp, -16
; RV64XTHEADBB-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64XTHEADBB-NEXT: negw a1, a0
; RV64XTHEADBB-NEXT: and a0, a0, a1
; RV64XTHEADBB-NEXT: lui a1, 30667
; RV64XTHEADBB-NEXT: addiw a1, a1, 1329
; RV64XTHEADBB-NEXT: call __muldi3
; RV64XTHEADBB-NEXT: srliw a0, a0, 27
; RV64XTHEADBB-NEXT: lui a1, %hi(.LCPI6_0)
; RV64XTHEADBB-NEXT: addi a1, a1, %lo(.LCPI6_0)
; RV64XTHEADBB-NEXT: add a0, a1, a0
; RV64XTHEADBB-NEXT: lbu a0, 0(a0)
; RV64XTHEADBB-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64XTHEADBB-NEXT: addi sp, sp, 16
; RV64XTHEADBB-NEXT: ret
; RV64XTHEADBB-NEXT: .LBB6_2:
; RV64XTHEADBB-NEXT: li a0, 32
; RV64XTHEADBB-NEXT: ret
; RV64ZBB-LABEL: cttz_i32:
; RV64ZBB: # %bb.0:
; RV64ZBB-NEXT: ctzw a0, a0
; RV64ZBB-NEXT: ret
%1 = call i32 @llvm.cttz.i32(i32 %a, i1 false)
ret i32 %1
}

```

```

define
  signext i32 @cttz_zero_undef_i32(i32 signext %a) nounwind {
; RV64I-LABEL: cttz_zero_undef_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  negw a1, a0
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  lui a1, 30667
; RV64I-NEXT:  addiw a1, a1, 1329
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:  srliw a0, a0, 27
; RV64I-NEXT:  lui a1, %hi(.LCPI7_0)
; RV64I-NEXT:  addi a1, a1, %lo(.LCPI7_0)
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  lbu a0, 0(a0)
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: cttz_zero_undef_i32:
; RV64XTHEADBB:  # %bb.0:
; RV64XTHEADBB-NEXT:  addi sp, sp, -16
; RV64XTHEADBB-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64XTHEADBB-NEXT:  negw a1, a0
; RV64XTHEADBB-NEXT:  and a0, a0, a1
; RV64XTHEADBB-NEXT:  lui a1, 30667
; RV64XTHEADBB-NEXT:  addiw a1, a1, 1329
; RV64XTHEADBB-NEXT:
;   call __muldi3
; RV64XTHEADBB-NEXT:  srliw a0, a0, 27
; RV64XTHEADBB-NEXT:  lui a1, %hi(.LCPI7_0)
; RV64XTHEADBB-NEXT:  addi a1, a1, %lo(.LCPI7_0)
; RV64XTHEADBB-NEXT:  add a0, a1, a0
; RV64XTHEADBB-NEXT:  lbu a0, 0(a0)
; RV64XTHEADBB-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64XTHEADBB-NEXT:  addi sp, sp, 16
; RV64XTHEADBB-NEXT:  ret
  %1 = call i32 @llvm.cttz.i32(i32 %a, i1 true)
  ret i32 %1
}

```

```

define signext i32 @findFirstSet_i32(i32 signext %a) nounwind {
; RV64I-LABEL: findFirstSet_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill

```

```

; RV64I-NEXT: sd s0, 0(sp) # 8-byte Folded Spill
; RV64I-NEXT: mv s0, a0
; RV64I-NEXT: negw a0, a0
; RV64I-NEXT: and a0, s0, a0
; RV64I-NEXT: lui a1, 30667
; RV64I-NEXT: addiw a1, a1, 1329
; RV64I-NEXT: call __muldi3
; RV64I-NEXT: srliw a0, a0, 27
; RV64I-NEXT: lui a1, %hi(.LCPI8_0)
; RV64I-NEXT: addi a1, a1, %lo(.LCPI8_0)
; RV64I-NEXT:
    add a0, a1, a0
; RV64I-NEXT: lbu a0, 0(a0)
; RV64I-NEXT: snez a1, s0
; RV64I-NEXT: addi a1, a1, -1
; RV64I-NEXT: or a0, a1, a0
; RV64I-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT: ld s0, 0(sp) # 8-byte Folded Reload
; RV64I-NEXT: addi sp, sp, 16
; RV64I-NEXT: ret
;
; RV64XTHEADBB-LABEL: findFirstSet_i32:
; RV64XTHEADBB:    # %bb.0:
; RV64XTHEADBB-NEXT: addi sp, sp, -16
; RV64XTHEADBB-NEXT: sd ra, 8(sp) # 8-byte Folded Spill
; RV64XTHEADBB-NEXT: sd s0, 0(sp) # 8-byte Folded Spill
; RV64XTHEADBB-NEXT: mv s0, a0
; RV64XTHEADBB-NEXT: negw a0, a0
; RV64XTHEADBB-NEXT: and a0, s0, a0
; RV64XTHEADBB-NEXT: lui a1, 30667
; RV64XTHEADBB-NEXT: addiw a1, a1, 1329
; RV64XTHEADBB-NEXT: call __muldi3
; RV64XTHEADBB-NEXT: srliw a0, a0, 27
; RV64XTHEADBB-NEXT: lui a1, %hi(.LCPI8_0)
; RV64XTHEADBB-NEXT: addi a1, a1, %lo(.LCPI8_0)
; RV64XTHEADBB-NEXT: add a0, a1, a0
; RV64XTHEADBB-NEXT: lbu a0, 0(a0)
; RV64XTHEADBB-NEXT:
    snez a1, s0
; RV64XTHEADBB-NEXT: addi a1, a1, -1
; RV64XTHEADBB-NEXT: or a0, a1, a0
; RV64XTHEADBB-NEXT: ld ra, 8(sp) # 8-byte Folded Reload
; RV64XTHEADBB-NEXT: ld s0, 0(sp) # 8-byte Folded Reload
; RV64XTHEADBB-NEXT: addi sp, sp, 16
; RV64XTHEADBB-NEXT: ret
%1 = call i32 @llvm.cttz.i32(i32 %a, i1 true)
%2 = icmp eq i32 %a, 0
%3 = select i1 %2, i32 -1, i32 %1

```

```
ret i32 %3
}
```

```
define signext i32 @ffs_i32(i32 signext %a) nounwind {
; RV64I-LABEL: ffs_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  sd s0, 0(sp) # 8-byte Folded Spill
; RV64I-NEXT:  mv s0, a0
; RV64I-NEXT:  negw a0, a0
; RV64I-NEXT:  and a0, s0, a0
; RV64I-NEXT:  lui a1, 30667
; RV64I-NEXT:  addiw a1, a1, 1329
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:  srliw a0, a0, 27
; RV64I-NEXT:  lui a1, %hi(.LCPI9_0)
; RV64I-NEXT:  addi a1, a1, %lo(.LCPI9_0)
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:
    lbu a0, 0(a0)
; RV64I-NEXT:  addi a0, a0, 1
; RV64I-NEXT:  seqz a1, s0
; RV64I-NEXT:  addi a1, a1, -1
; RV64I-NEXT:  and a0, a1, a0
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  ld s0, 0(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: ffs_i32:
; RV64XTHEADBB:      # %bb.0:
; RV64XTHEADBB-NEXT:  addi sp, sp, -16
; RV64XTHEADBB-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64XTHEADBB-NEXT:  sd s0, 0(sp) # 8-byte Folded Spill
; RV64XTHEADBB-NEXT:  mv s0, a0
; RV64XTHEADBB-NEXT:  negw a0, a0
; RV64XTHEADBB-NEXT:  and a0, s0, a0
; RV64XTHEADBB-NEXT:  lui a1, 30667
; RV64XTHEADBB-NEXT:  addiw a1, a1, 1329
; RV64XTHEADBB-NEXT:  call __muldi3
; RV64XTHEADBB-NEXT:  srliw a0, a0, 27
; RV64XTHEADBB-NEXT:  lui a1, %hi(.LCPI9_0)
; RV64XTHEADBB-NEXT:  addi a1, a1, %lo(.LCPI9_0)
; RV64XTHEADBB-NEXT:  add a0, a1, a0
;
}
```

```

RV64XTHEADBB-NEXT:  lbu a0, 0(a0)
; RV64XTHEADBB-NEXT:  addi a0, a0, 1
; RV64XTHEADBB-NEXT:  seqz a1, s0
; RV64XTHEADBB-NEXT:  addi a1, a1, -1
; RV64XTHEADBB-NEXT:  and a0, a1, a0
; RV64XTHEADBB-NEXT:  slli a0, a0, 32
; RV64XTHEADBB-NEXT:  srli a0, a0, 32
; RV64XTHEADBB-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64XTHEADBB-NEXT:  ld s0, 0(sp) # 8-byte Folded Reload
; RV64XTHEADBB-NEXT:  addi sp, sp, 16
; RV64XTHEADBB-NEXT:  ret
%1 = call i32 @llvm.cttz.i32(i32 %a, i1 true)
%2 = add i32 %1, 1
%3 = icmp eq i32 %a, 0
%4 = select i1 %3, i32 0, i32 %2
ret i32 %4
}

```

```

declare i64 @llvm.cttz.i64(i64, i1)

```

```

define i64 @cttz_i64(i64 %a) nounwind {
; RV64I-LABEL: cttz_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  beqz a0, .LBB10_2
; RV64I-NEXT:  # %bb.1: # %cond.false
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  neg a1, a0
; RV64I-NEXT:  and a0, a0, a1
; RV64I-NEXT:  lui a1, %hi(.LCPI10_0)
; RV64I-NEXT:
    ld a1, %lo(.LCPI10_0)(a1)
; RV64I-NEXT:  call __muldi3
; RV64I-NEXT:  srli a0, a0, 58
; RV64I-NEXT:  lui a1, %hi(.LCPI10_1)
; RV64I-NEXT:  addi a1, a1, %lo(.LCPI10_1)
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  lbu a0, 0(a0)
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
; RV64I-NEXT: .LBB10_2:
; RV64I-NEXT:  li a0, 64
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: cttz_i64:
; RV64XTHEADBB:  # %bb.0:
; RV64XTHEADBB-NEXT:  beqz a0, .LBB10_2

```



```

; RV64XTHEADBB-NEXT: # %bb.1: # %cond.false
; RV64XTHEADBB-NEXT:  addi a1, a0, -1
; RV64XTHEADBB-NEXT:  not a0, a0
; RV64XTHEADBB-NEXT:  and a0, a0, a1
; RV64XTHEADBB-NEXT:  th.ff1 a0, a0
; RV64XTHEADBB-NEXT:  li a1, 64
; RV64XTHEADBB-NEXT:  sub a0, a1, a0
; RV64XTHEADBB-NEXT:  ret
; RV64XTHEADBB-NEXT: .LBB10_2:
; RV64XTHEADBB-NEXT:  li a0, 64
; RV64XTHEADBB-NEXT:  ret
%1 = call i64 @llvm.cttz.i64(i64 %a, i1 false)
ret i64 %1
}

```

```

define signext
i32 @sextb_i32(i32 signext %a) nounwind {
; RV64I-LABEL: sextb_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 56
; RV64I-NEXT:  srai a0, a0, 56
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: sextb_i32:
; RV64XTHEADBB:  # %bb.0:
; RV64XTHEADBB-NEXT:  th.ext a0, a0, 7, 0
; RV64XTHEADBB-NEXT:  ret
%shl = shl i32 %a, 24
%shr = ashr exact i32 %shl, 24
ret i32 %shr
}

```

```

define i64 @sextb_i64(i64 %a) nounwind {
; RV64I-LABEL: sextb_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 56
; RV64I-NEXT:  srai a0, a0, 56
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: sextb_i64:
; RV64XTHEADBB:  # %bb.0:
; RV64XTHEADBB-NEXT:  th.ext a0, a0, 7, 0
; RV64XTHEADBB-NEXT:  ret
%shl = shl i64 %a, 56
%shr = ashr exact i64 %shl, 56
ret i64 %shr
}

```

```

define signext i32 @sexth_i32(i32 signext %a) nounwind {
; RV64I-LABEL: sexth_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srai a0, a0, 48
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL:
sexth_i32:
; RV64XTHEADBB:      # %bb.0:
; RV64XTHEADBB-NEXT:  th.ext a0, a0, 15, 0
; RV64XTHEADBB-NEXT:  ret
%shl = shl i32 %a, 16
%shr = ashr exact i32 %shl, 16
ret i32 %shr
}

```

```

define signext i32 @no_sexth_i32(i32 signext %a) nounwind {
; RV64I-LABEL: no_sexth_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 17
; RV64I-NEXT:  sraiw a0, a0, 16
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: no_sexth_i32:
; RV64XTHEADBB:      # %bb.0:
; RV64XTHEADBB-NEXT:  slli a0, a0, 17
; RV64XTHEADBB-NEXT:  sraiw a0, a0, 16
; RV64XTHEADBB-NEXT:  ret
%shl = shl i32 %a, 17
%shr = ashr exact i32 %shl, 16
ret i32 %shr
}

```

```

define i64 @sexth_i64(i64 %a) nounwind {
; RV64I-LABEL: sexth_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srai a0, a0, 48
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: sexth_i64:
; RV64XTHEADBB:      # %bb.0:
; RV64XTHEADBB-NEXT:  th.ext a0, a0, 15, 0
; RV64XTHEADBB-NEXT:  ret
%shl = shl i64 %a, 48
%shr = ashr

```

```

exact i64 %shl, 48
ret i64 %shr
}

```

```

define i64 @no_s sixth_i64(i64 %a) nounwind {
; RV64I-LABEL: no_s sixth_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 49
; RV64I-NEXT:  srai a0, a0, 48
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: no_s sixth_i64:
; RV64XTHEADBB:  # %bb.0:
; RV64XTHEADBB-NEXT:  slli a0, a0, 49
; RV64XTHEADBB-NEXT:  srai a0, a0, 48
; RV64XTHEADBB-NEXT:  ret
%shl = shl i64 %a, 49
%shr = ashr exact i64 %shl, 48
ret i64 %shr
}

```

```

define i32 @z sixth_i32(i32 %a) nounwind {
; RV64I-LABEL: z sixth_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srli a0, a0, 48
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: z sixth_i32:
; RV64XTHEADBB:  # %bb.0:
; RV64XTHEADBB-NEXT:  th.extu a0, a0, 15, 0
; RV64XTHEADBB-NEXT:  ret
%and = and i32 %a, 65535
ret i32 %and
}

```

```

define i64 @z sixth_i64(i64 %a) nounwind {
; RV64I-LABEL: z sixth_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srli a0, a0, 48
; RV64I-NEXT:
ret
;
; RV64XTHEADBB-LABEL: z sixth_i64:
; RV64XTHEADBB:  # %bb.0:
; RV64XTHEADBB-NEXT:  th.extu a0, a0, 15, 0
; RV64XTHEADBB-NEXT:  ret

```

```

%and = and i64 %a, 65535
ret i64 %and
}

define i64 @zext_bf_i64(i64 %a) nounwind {
; RV64I-LABEL: zext_bf_i64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 47
; RV64I-NEXT:  srli a0, a0, 48
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: zext_bf_i64:
; RV64XTHEADBB:  # %bb.0:
; RV64XTHEADBB-NEXT:  th.extu a0, a0, 16, 1
; RV64XTHEADBB-NEXT:  ret
%1 = lshr i64 %a, 1
%and = and i64 %1, 65535
ret i64 %and
}

define i64 @zext_i64_srliw(i64 %a) nounwind {
; RV64I-LABEL: zext_i64_srliw:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srliw a0, a0, 16
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: zext_i64_srliw:
; RV64XTHEADBB:  # %bb.0:
; RV64XTHEADBB-NEXT:  srliw a0, a0, 16
; RV64XTHEADBB-NEXT:  ret
%1 = lshr i64 %a, 16
%and = and i64 %1, 65535
ret i64 %and
}

declare i32 @llvm.bswap.i32(i32)

define signext
i32 @bswap_i32(i32 signext %a) nounwind {
; RV64I-LABEL: bswap_i32:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srliw a1, a0, 8
; RV64I-NEXT:  lui a2, 16
; RV64I-NEXT:  addiw a2, a2, -256
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  srliw a3, a0, 24
; RV64I-NEXT:  or a1, a1, a3
; RV64I-NEXT:  and a2, a0, a2

```

```

; RV64I-NEXT:  slliw a2, a2, 8
; RV64I-NEXT:  slliw a0, a0, 24
; RV64I-NEXT:  or a0, a0, a2
; RV64I-NEXT:  or a0, a0, a1
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: bswap_i32:
; RV64XTHEADBB:      # %bb.0:
; RV64XTHEADBB-NEXT:  th.revw a0, a0
; RV64XTHEADBB-NEXT:  ret
%1 = tail call i32 @llvm.bswap.i32(i32 %a)
ret i32 %1
}

; Similar to bswap_i32 but the result is not sign extended.
define void @bswap_i32_nosext(i32 signext %a, ptr %x) nounwind {
; RV64I-LABEL: bswap_i32_nosext:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srlw a2, a0, 8
; RV64I-NEXT:  lui a3, 16
; RV64I-NEXT:  addi a3, a3, -256
; RV64I-NEXT:  and a2, a2, a3
; RV64I-NEXT:  srlw
a4, a0, 24
; RV64I-NEXT:  or a2, a2, a4
; RV64I-NEXT:  and a3, a0, a3
; RV64I-NEXT:  slli a3, a3, 8
; RV64I-NEXT:  slli a0, a0, 24
; RV64I-NEXT:  or a0, a0, a3
; RV64I-NEXT:  or a0, a0, a2
; RV64I-NEXT:  sw a0, 0(a1)
; RV64I-NEXT:  ret
;
; RV64XTHEADBB-LABEL: bswap_i32_nosext:
; RV64XTHEADBB:      # %bb.0:
; RV64XTHEADBB-NEXT:  th.revw a0, a0
; RV64XTHEADBB-NEXT:  sw a0, 0(a1)
; RV64XTHEADBB-NEXT:  ret
%1 = tail call i32 @llvm.bswap.i32(i32 %a)
store i32 %1, ptr %x
ret void
}

declare i64 @llvm.bswap.i64(i64)

define i64 @bswap_i64(i64 %a) {
; RV64I-LABEL: bswap_i64:
; RV64I:      # %bb.0:

```

```

; RV64I-NEXT: srli a1, a0, 40
; RV64I-NEXT: lui a2, 16
; RV64I-NEXT: addiw a2, a2, -256
; RV64I-NEXT: and a1, a1, a2
; RV64I-NEXT: srli a3, a0, 56
; RV64I-NEXT: or a1, a1, a3
; RV64I-NEXT: srli a3, a0, 24
; RV64I-NEXT: lui a4, 4080
; RV64I-NEXT: and a3, a3, a4
; RV64I-NEXT: srli a5, a0, 8
; RV64I-NEXT: srliw a5, a5, 24
; RV64I-NEXT: slli
a5, a5, 24
; RV64I-NEXT: or a3, a5, a3
; RV64I-NEXT: or a1, a3, a1
; RV64I-NEXT: and a4, a0, a4
; RV64I-NEXT: slli a4, a4, 24
; RV64I-NEXT: srliw a3, a0, 24
; RV64I-NEXT: slli a3, a3, 32
; RV64I-NEXT: or a3, a4, a3
; RV64I-NEXT: and a2, a0, a2
; RV64I-NEXT: slli a2, a2, 40
; RV64I-NEXT: slli a0, a0, 56
; RV64I-NEXT: or a0, a0, a2
; RV64I-NEXT: or a0, a0, a3
; RV64I-NEXT: or a0, a0, a1
; RV64I-NEXT: ret
;
; RV64XTHEADBB-LABEL: bswap_i64:
; RV64XTHEADBB:    # %bb.0:
; RV64XTHEADBB-NEXT:  th.rev a0, a0
; RV64XTHEADBB-NEXT:  ret
%1 = call i64 @llvm.bswap.i64(i64 %a)
ret i64 %1
}
; RUN: llc < %s -mtriple=s390x-linux-gnu -mcpu=zEC12 -verify-machineinstrs | FileCheck %s
;
; Test that early if conversion produces LOCR with operands of the right
; register classes.

define void @autogen_SD4739(ptr) {
; CHECK-NOT: Expected a GR32Bit register, but got a GRX32Bit register
BB:
%L34 = load i8, ptr %0
%Cmp56 = icmp sgt i8 undef, %L34
br label %CF246

CF246:                                ; preds = %CF246, %BB

```

```
%S1163 = select i1 %Cmp56, i8 %L34, i8 undef
br i1 undef, label %CF246, label %CF248
```

```
CF248:                                ; preds = %CF248, %CF246
store i8 %S1163, ptr %0
br label %CF248
}
```

```
; RUN: llc -march=hexagon -hexagon-hvx-widen=32 < %s | FileCheck %s
```

```
; Truncating a type-to-be-widened to a legal type (v8i8).
```

```
; Check that this compiles successfully.
```

```
; CHECK-LABEL: f0:
```

```
; CHECK: dealloc_return
```

```
target datalayout = "e-m:e-p:32:32:32-a:0-n16:32-i64:64:64-i32:32:32-i16:16:16-i1:8:8-f32:32:32-f64:64:64-
v32:32:32-v64:64:64-v512:512:512-v1024:1024:1024-v2048:2048:2048"
target triple = "hexagon"
```

```
define dlllexport void @f0(ptr %a0) local_unnamed_addr #0 {
b0:
%v0 = load i8, ptr undef, align 1
%v1 = zext i8 %v0 to i16
%v2 = add i16 0, %v1
%v3 = icmp sgt i16 %v2, 1
%v4 = select i1 %v3, i16 %v2, i16 1
%v5 = udiv i16 -32768, %v4
%v6 = zext i16 %v5 to i32
%v7 = insertelement <8 x i32> undef, i32 %v6, i32 0
%v8 = shufflevector <8 x i32> %v7, <8 x i32> undef, <8 x i32> zeroinitializer
%v9 = load <8 x i16>, ptr undef, align 2
%v10 = sext <8 x i16> %v9 to <8 x i32>
%v11 = mul nsw <8 x i32> %v8, %v10
%v12 = add nsw <8 x i32> %v11, <i32 16384, i32 16384, i32 16384,
i32 16384, i32 16384, i32 16384, i32 16384, i32 16384>
%v13 = lshr <8 x i32> %v12, <i32 15, i32 15, i32 15, i32 15, i32 15, i32 15, i32 15, i32 15>
%v14 = trunc <8 x i32> %v13 to <8 x i8>
%v15 = getelementptr inbounds i8, ptr %a0, i32 undef
store <8 x i8> %v14, ptr %v15, align 1
ret void
}
```

```
attributes #0 = { "target-features"="+hvx,+hvx-length128b" }
```

```
; RUN: llc -O3 -mtriple=powerpc-unknown-linux-gnu -mcpu=e500 -mattr=spe < %s | FileCheck %s
```

```
; PowerPC SPE is a rare in-tree target that has the FP_TO_SINT node marked
```

```
; as Legal.
```

```
; Verify that fptosi(42.1) isn't simplified when the rounding mode is
```

```

; unknown.
; Verify that no gross errors happen.
; CHECK-LABEL: @f20
; COMMON: cfdctsiz
define i32 @f20(double %a) strictfp {
entry:
  %result = call i32 @llvm.experimental.constrained.fptosi.i32.f64(double 42.1,
                                metadata !"fpexcept.strict")
                                strictfp
  ret i32 %result
}

```

```

@llvm.fp.env = thread_local global i8 zeroinitializer, section "llvm.metadata"
declare i32 @llvm.experimental.constrained.fptosi.i32.f64(double, metadata)
Copyright (c) 2019 Intel Corporation. All rights reserved.

```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

; NOTE: Assertions have been autogenerated by utils/update\_analyze\_test\_checks.py

; RUN: opt < %s -passes="print<cost-model>" 2>&1 -disable-output -mtriple=x86\_64-apple-macosx10.8.0 -mattr=+avx2 | FileCheck %s --check-prefixes=VEC256,AVX

; RUN: opt < %s -passes="print<cost-model>" 2>&1 -disable-output -mtriple=x86\_64-apple-macosx10.8.0 -mattr=+avx512vl,+prefer-256-bit | FileCheck %s --check-prefixes=VEC256,AVX512VL256

; RUN: opt < %s -passes="print<cost-model>" 2>&1 -disable-output -mtriple=x86\_64-apple-macosx10.8.0 -mattr=+avx512vl,+prefer-256-bit | FileCheck %s --check-prefixes=AVX512VL512

; RUN: opt < %s -passes="print<cost-model>" 2>&1 -disable-output -mtriple=x86\_64-apple-macosx10.8.0 -mattr=+avx512vl,+avx512bw,+avx512dq,+prefer-256-bit | FileCheck %s --check-prefixes=VEC256,SKX256

; RUN: opt < %s -passes="print<cost-model>" 2>&1 -disable-output -mtriple=x86\_64-apple-macosx10.8.0 -mattr=+avx512vl,+avx512bw,+avx512dq,-prefer-256-bit | FileCheck %s --check-prefixes=SKX512



```

define
void @zext256() "min-legal-vector-width"="256" {
; AVX-LABEL: 'zext256'
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %A = zext <8 x i16> undef to <8 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %B = zext <8 x i32> undef to <8 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %C = zext <16 x i8> undef to <16 x i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %D = zext <16 x i16> undef to <16 x
i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %E = zext <32 x i8> undef to <32 x i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL256-LABEL: 'zext256'
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %A = zext <8 x i16> undef to
<8 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %B = zext <8 x i32> undef to
<8 x i64>
; AVX512VL256-NEXT: Cost
Model: Found an estimated cost of 2 for instruction: %C = zext <16 x i8> undef to <16 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %D = zext <16 x i16> undef to
<16 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %E = zext <32 x i8> undef to
<32 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL512-LABEL: 'zext256'
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = zext <8 x i16> undef to
<8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = zext <8 x i32> undef to
<8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = zext <16 x i8> undef to
<16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = zext <16 x i16> undef to
<16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction:
%E = zext <32 x i8> undef to <32 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX256-LABEL: 'zext256'
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %A = zext <8 x i16> undef to <8 x
i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %B = zext <8 x i32> undef to <8 x
i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %C = zext <16 x i8> undef to <16 x
i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %D = zext <16 x i16> undef to <16 x
i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %E = zext <32 x i8> undef to <32 x
i16>

```

```

; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX512-LABEL: 'zext256'
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = zext <8 x i16> undef to <8 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = zext <8 x i32> undef to <8 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = zext <16 x i8> undef to <16 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = zext <16 x i16> undef to <16 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = zext <32 x i8> undef to <32 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
%A = zext <8 x i16> undef to <8 x i64>
%B = zext <8 x i32> undef to <8 x i64>
%C = zext <16 x i8> undef to <16 x i32>
%D = zext <16 x i16> undef to <16 x i32>
%E = zext <32 x i8> undef to <32 x i16>
ret void
}

define void @zext512() "min-legal-vector-width"="512" {
; AVX-LABEL: 'zext512'
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %A = zext <8 x i16> undef to <8 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %B = zext <8 x i32> undef to <8 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %C = zext <16 x i8> undef to <16 x i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %D = zext <16 x i16> undef to <16 x i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %E = zext <32 x i8> undef to <32 x i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL256-LABEL: 'zext512'
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = zext <8 x i16> undef to <8 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = zext <8 x i32> undef to <8 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = zext <16 x i8> undef to <16 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = zext <16 x i16> undef to <16 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %E = zext <32 x i8> undef to <32 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL512-LABEL: 'zext512'

```

```

; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = zext <8 x i16> undef to
<8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = zext <8 x i32> undef to
<8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = zext <16 x i8> undef to
<16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = zext <16 x i16> undef to
<16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %E = zext <32 x i8> undef to
<32 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX256-LABEL: 'zext512'
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = zext <8 x i16> undef to <8
x i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = zext <8 x i32> undef to <8 x
i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = zext <16 x i8> undef to <16 x
i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = zext <16 x i16> undef to <16 x
i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = zext <32 x i8> undef to <32 x
i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX512-LABEL: 'zext512'
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = zext <8 x i16> undef to <8 x
i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = zext <8 x i32> undef to <8 x
i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = zext <16 x i8> undef to <16 x
i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = zext <16
x i16> undef to <16 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = zext <32 x i8> undef to <32 x
i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
%A = zext <8 x i16> undef to <8 x i64>
%B = zext <8 x i32> undef to <8 x i64>
%C = zext <16 x i8> undef to <16 x i32>
%D = zext <16 x i16> undef to <16 x i32>
%E = zext <32 x i8> undef to <32 x i16>
ret void
}

define void @sext256() "min-legal-vector-width"="256" {
; AVX-LABEL: 'sext256'

```

```

; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %A = sext <8 x i8> undef to <8 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %B = sext <8 x i16> undef to <8 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %C = sext <8 x i32> undef to <8 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %D = sext <16 x i8> undef to <16 x i32>
; AVX-NEXT: Cost Model: Found
an estimated cost of 3 for instruction: %E = sext <16 x i16> undef to <16 x i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %F = sext <32 x i8> undef to <32 x i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL256-LABEL: 'sext256'
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %A = sext <8 x i8> undef to <8
x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %B = sext <8 x i16> undef to
<8 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %C = sext <8 x i32> undef to
<8 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %D = sext <16 x i8> undef to
<16 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %E = sext <16 x i16> undef to
<16 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %F = sext <32 x i8> undef
to <32 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL512-LABEL: 'sext256'
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = sext <8 x i8> undef to <8
x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to
<8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to
<8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = sext <16 x i8> undef to
<16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = sext <16 x i16> undef to
<16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %F = sext <32 x i8> undef to
<32 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX256-LABEL: 'sext256'
; SKX256-NEXT: Cost Model: Found an
estimated cost of 2 for instruction: %A = sext <8 x i8> undef to <8 x i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %B = sext <8 x i16> undef to <8 x
i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %C = sext <8 x i32> undef to <8 x
i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %D = sext <16 x i8> undef to <16 x
i32>

```

```
; SKX256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %E = sext <16 x i16> undef to <16 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %F = sext <32 x i8> undef to <32 x i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX512-LABEL: 'sext256'
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = sext <8 x i8> undef to <8 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to <8 x i64>
; SKX512-NEXT: Cost
Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to <8 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = sext <16 x i8> undef to <16 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = sext <16 x i16> undef to <16 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %F = sext <32 x i8> undef to <32 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
%A = sext <8 x i8> undef to <8 x i64>
%B = sext <8 x i16> undef to <8 x i64>
%C = sext <8 x i32> undef to <8 x i64>
%D = sext <16 x i8> undef to <16 x i32>
%E = sext <16 x i16> undef to <16 x i32>
%F = sext <32 x i8> undef to <32 x i16>
ret void
}
```

```
define void @sext512() "min-legal-vector-width"="512" {
; AVX-LABEL: 'sext512'
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %A = sext <8 x i8> undef to <8 x i64>
; AVX-NEXT: Cost Model: Found
an estimated cost of 4 for instruction: %B = sext <8 x i16> undef to <8 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %C = sext <8 x i32> undef to <8 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %D = sext <16 x i8> undef to <16 x i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %E = sext <16 x i16> undef to <16 x i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %F = sext <32 x i8> undef to <32 x i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL256-LABEL: 'sext512'
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = sext <8 x i8> undef to <8 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to <8 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to <8 x i64>
; AVX512VL256-NEXT:
```

Cost Model: Found an estimated cost of 1 for instruction: %D = sext <16 x i8> undef to <16 x i32>

; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = sext <16 x i16> undef to <16 x i32>

; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %F = sext <32 x i8> undef to <32 x i16>

; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void

;

; AVX512VL512-LABEL: 'sext512'

; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = sext <8 x i8> undef to <8 x i64>

; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to <8 x i64>

; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to <8 x i64>

; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = sext <16 x i8> undef to <16 x i32>

; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = sext <16 x i16> undef to <16 x i32>

; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %F = sext <32 x i8> undef to <32 x i16>

; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void

;

; SKX256-LABEL: 'sext512'

; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = sext <8 x i8> undef to <8 x i64>

; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to <8 x i64>

; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to <8 x i64>

; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = sext <16 x i8> undef to <16 x i32>

; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = sext <16 x i16> undef to <16 x i32>

; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %F = sext <32 x i8> undef to <32 x i16>

; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void

;

; SKX512-LABEL: 'sext512'

; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = sext <8 x i8> undef to <8 x i64>

; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to <8 x i64>

; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to <8 x i64>

; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = sext <16 x i8> undef to <16 x i32>

; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = sext <16 x i16> undef to <16 x i32>

; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %F = sext <32 x i8> undef to <32 x i16>

```

i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
%A = sext <8 x i8> undef to <8 x i64>
%B = sext <8 x i16> undef to <8 x i64>
%C = sext <8 x i32> undef to <8 x i64>
%D = sext <16 x i8>
undef to <16 x i32>
%E = sext <16 x i16> undef to <16 x i32>
%F = sext <32 x i8> undef to <32 x i16>
ret void
}

define void @trunc256() "min-legal-vector-width"="256" {
; VEC256-LABEL: 'trunc256'
; VEC256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %A = trunc <8 x i64> undef to <8 x
i32>
; VEC256-NEXT: Cost Model: Found an estimated cost of 10 for instruction: %B = trunc <8 x i64> undef to <8 x
i16>
; VEC256-NEXT: Cost Model: Found an estimated cost of 8 for instruction: %C = trunc <8 x i64> undef to <8 x
i8>
; VEC256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %D = trunc <16 x i32> undef to <16 x
i16>
; VEC256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %E = trunc <16 x i32> undef to <16 x
i8>
; VEC256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %F = trunc <32 x i16> undef to <32 x
i8>
; VEC256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL512-LABEL: 'trunc256'
;
AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = trunc <8 x i64> undef to
<8 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %B = trunc <8 x i64> undef to
<8 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %C = trunc <8 x i64> undef to
<8 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %D = trunc <16 x i32> undef
to <16 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %E = trunc <16 x i32> undef to
<16 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 8 for instruction: %F = trunc <32 x i16> undef to
<32 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX512-LABEL: 'trunc256'
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = trunc <8 x i64> undef to <8 x
i32>

```

```

; SKX512-NEXT: Cost Model: Found an estimated
cost of 2 for instruction: %B = trunc <8 x i64> undef to <8 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %C = trunc <8 x i64> undef to <8 x
i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %D = trunc <16 x i32> undef to <16 x
i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %E = trunc <16 x i32> undef to <16 x
i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %F = trunc <32 x i16> undef to <32 x
i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
%A = trunc <8 x i64> undef to <8 x i32>
%B = trunc <8 x i64> undef to <8 x i16>
%C = trunc <8 x i64> undef to <8 x i8>
%D = trunc <16 x i32> undef to <16 x i16>
%E = trunc <16 x i32> undef to <16 x i8>
%F = trunc <32 x i16> undef to <32 x i8>
ret void
}

define i32 @zext256_vXi1() "min-legal-vector-width"="256" {
; AVX-LABEL: 'zext256_vXi1'
; AVX-NEXT: Cost Model: Found
an estimated cost of 1 for instruction: %V2i64 = zext <2 x i1> undef to <2 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V4i64 = zext <4 x i1> undef to <4 x
i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V8i64 = zext <8 x i1> undef to <8 x
i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i32 = zext <2 x i1> undef to <2 x
i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i32 = zext <4 x i1> undef to <4 x
i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V8i32 = zext <8 x i1> undef to <8 x
i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V16i32 = zext <16 x i1> undef to <16 x
i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i16 = zext <2 x i1> undef to <2 x
i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i16 = zext <4 x i1> undef to <4 x
i16>
; AVX-NEXT:
Cost Model: Found an estimated cost of 1 for instruction: %V8i16 = zext <8 x i1> undef to <8 x i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i16 = zext <16 x i1> undef to <16 x
i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V32i16 = zext <32 x i1> undef to <32 x
i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i8 = zext <2 x i1> undef to <2 x i8>

```



```

; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i8 = zext <4 x i1> undef to <4 x i8>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i8 = zext <8 x i1> undef to <8 x i8>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i8 = zext <16 x i1> undef to <16 x i8>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V32i8 = zext <32 x i1> undef to <32 x i8>
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V64i8 = zext <64 x i1> undef to <64 x i8>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; AVX512VL256-LABEL: 'zext256_vXi1'
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = zext <2 x i1> undef to <2 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = zext <4 x i1> undef to <4 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V8i64 = zext <8 x i1> undef to <8 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i32 = zext <2 x i1> undef to <2 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = zext <4 x i1> undef to <4 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = zext <8 x i1> undef to <8 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V16i32 = zext <16 x i1> undef to <16 x i32>
; AVX512VL256-NEXT:
  Cost Model: Found an estimated cost of 5 for instruction: %V2i16 = zext <2 x i1> undef to <2 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V4i16 = zext <4 x i1> undef to <4 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V8i16 = zext <8 x i1> undef to <8 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 12 for instruction: %V16i16 = zext <16 x i1> undef to <16 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 24 for instruction: %V32i16 = zext <32 x i1> undef to <32 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 6 for instruction: %V2i8 = zext <2 x i1> undef to <2 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 6 for instruction: %V4i8 = zext <4 x i1> undef to <4 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 6 for instruction: %V8i8 = zext <8 x i1> undef to <8 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 12 for instruction: %V16i8 = zext <16 x i1> undef to <16 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 25 for instruction: %V32i8 = zext <32 x i1> undef to <32 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 50 for instruction: %V64i8 = zext <64 x i1> undef to <64 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef

```

```

;
; AVX512VL512-LABEL: 'zext256_vXi1'
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = zext <2 x i1> undef
to <2 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = zext <4 x i1> undef
to <4 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i64 = zext <8 x i1> undef
to <8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i32 = zext <2 x i1> undef
to <2 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of
2 for instruction: %V4i32 = zext <4 x i1> undef to <4 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = zext <8 x i1> undef
to <8 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i32 = zext <16 x i1>
undef to <16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V2i16 = zext <2 x i1> undef
to <2 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V4i16 = zext <4 x i1> undef
to <4 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V8i16 = zext <8 x i1> undef
to <8 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V16i16 = zext <16 x i1>
undef to <16 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 9 for instruction: %V32i16 = zext <32 x i1>
undef to <32 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V2i8
= zext <2 x i1> undef to <2 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V4i8 = zext <4 x i1> undef to
<4 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V8i8 = zext <8 x i1> undef to
<8 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V16i8 = zext <16 x i1> undef
to <16 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 9 for instruction: %V32i8 = zext <32 x i1> undef
to <32 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 19 for instruction: %V64i8 = zext <64 x i1>
undef to <64 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; SKX256-LABEL: 'zext256_vXi1'
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = zext <2 x i1> undef to <2 x
i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = zext <4 x i1> undef to <4 x
i64>
;
; SKX256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V8i64 = zext <8 x i1> undef to <8 x
i64>

```

```

; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i32 = zext <2 x i1> undef to <2 x
i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = zext <4 x i1> undef to <4 x
i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = zext <8 x i1> undef to <8 x
i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V16i32 = zext <16 x i1> undef to
<16 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i16 = zext <2 x i1> undef to <2 x
i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i16 = zext <4 x i1> undef to <4 x
i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i16 = zext <8 x i1> undef to <8 x
i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction:
%V16i16 = zext <16 x i1> undef to <16 x i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V32i16 = zext <32 x i1> undef to
<32 x i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i8 = zext <2 x i1> undef to <2 x
i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i8 = zext <4 x i1> undef to <4 x
i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i8 = zext <8 x i1> undef to <8 x
i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i8 = zext <16 x i1> undef to <16
x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V32i8 = zext <32 x i1> undef to <32
x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V64i8 = zext <64 x i1> undef to <64
x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; SKX512-LABEL: 'zext256_vXi1'
; SKX512-NEXT: Cost
Model: Found an estimated cost of 2 for instruction: %V2i64 = zext <2 x i1> undef to <2 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = zext <4 x i1> undef to <4 x
i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i64 = zext <8 x i1> undef to <8 x
i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i32 = zext <2 x i1> undef to <2 x
i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = zext <4 x i1> undef to <4 x
i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = zext <8 x i1> undef to <8 x
i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i32 = zext <16 x i1> undef to
<16 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i16 = zext <2 x i1> undef to <2 x

```

```

i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i16 = zext
<4 x i1> undef to <4 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i16 = zext <8 x i1> undef to <8 x
i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i16 = zext <16 x i1> undef to
<16 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V32i16 = zext <32 x i1> undef to
<32 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i8 = zext <2 x i1> undef to <2 x
i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i8 = zext <4 x i1> undef to <4 x
i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i8 = zext <8 x i1> undef to <8 x
i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i8 = zext <16 x i1> undef to <16
x i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V32i8 = zext <32 x i1> undef to <32
x i8>
; SKX512-NEXT: Cost Model: Found an estimated
cost of 2 for instruction: %V64i8 = zext <64 x i1> undef to <64 x i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
%V2i64 = zext <2 x i1> undef to <2 x i64>
%V4i64 = zext <4 x i1> undef to <4 x i64>
%V8i64 = zext <8 x i1> undef to <8 x i64>

%V2i32 = zext <2 x i1> undef to <2 x i32>
%V4i32 = zext <4 x i1> undef to <4 x i32>
%V8i32 = zext <8 x i1> undef to <8 x i32>
%V16i32 = zext <16 x i1> undef to <16 x i32>

%V2i16 = zext <2 x i1> undef to <2 x i16>
%V4i16 = zext <4 x i1> undef to <4 x i16>
%V8i16 = zext <8 x i1> undef to <8 x i16>
%V16i16 = zext <16 x i1> undef to <16 x i16>
%V32i16 = zext <32 x i1> undef to <32 x i16>

%V2i8 = zext <2 x i1> undef to <2 x i8>
%V4i8 = zext <4 x i1> undef to <4 x i8>
%V8i8 = zext <8 x i1> undef to <8 x i8>
%V16i8 = zext <16 x i1> undef to <16 x i8>
%V32i8 = zext <32 x i1> undef to <32 x i8>
%V64i8 = zext <64 x i1> undef to <64 x i8>

ret i32 undef
}

```

```

define i32 @sext256_vXi1()
  "min-legal-vector-width"="256" {
; AVX-LABEL: 'sext256_vXi1'
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %i64 = sext i1 undef to i64
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = sext <2 x i1> undef to <2 x
i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V4i64 = sext <4 x i1> undef to <4 x
i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V8i64 = sext <8 x i1> undef to <8 x
i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %i32 = sext i1 undef to i32
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i32 = sext <2 x i1> undef to <2 x
i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = sext <4 x i1> undef to <4 x
i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V8i32 = sext <8 x i1> undef to <8 x
i32>
; AVX-NEXT: Cost Model: Found an estimated cost
of 4 for instruction: %V16i32 = sext <16 x i1> undef to <16 x i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %i16 = sext i1 undef to i16
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i16 = sext <2 x i1> undef to <2 x
i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i16 = sext <4 x i1> undef to <4 x
i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i16 = sext <8 x i1> undef to <8 x
i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i16 = sext <16 x i1> undef to <16 x
i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V32i16 = sext <32 x i1> undef to <32 x
i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %i8 = sext i1 undef to i8
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i8 = sext <2 x i1> undef to <2 x i8>
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for
instruction: %V4i8 = sext <4 x i1> undef to <4 x i8>
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i8 = sext <8 x i1> undef to <8 x i8>
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i8 = sext <16 x i1> undef to <16 x
i8>
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V32i8 = sext <32 x i1> undef to <32 x
i8>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V64i8 = sext <64 x i1> undef to <64 x
i8>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; AVX512VL256-LABEL: 'sext256_vXi1'
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %i64 = sext i1 undef to i64
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i64 = sext <2 x i1> undef
to <2 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i64 = sext <4 x i1> undef

```

```

to <4 x i64>
; AVX512VL256-NEXT:
  Cost Model: Found an estimated cost of 3 for instruction: %V8i64 = sext <8 x i1> undef to <8 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I32 = sext i1 undef to i32
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i32 = sext <2 x i1> undef
to <2 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i32 = sext <4 x i1> undef
to <4 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i32 = sext <8 x i1> undef
to <8 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i32 = sext <16 x i1>
undef to <16 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I16 = sext i1 undef to i16
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V2i16 = sext <2 x i1> undef
to <2 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 4 for instruction:
  %V4i16 = sext <4 x i1> undef to <4 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V8i16 = sext <8 x i1> undef
to <8 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 10 for instruction: %V16i16 = sext <16 x i1>
undef to <16 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 20 for instruction: %V32i16 = sext <32 x i1>
undef to <32 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I8 = sext i1 undef to i8
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V2i8 = sext <2 x i1> undef to
<2 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V4i8 = sext <4 x i1> undef to
<4 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V8i8 = sext <8 x i1> undef to
<8 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 10 for instruction: %V16i8 = sext <16 x i1>
undef to <16 x i8>
;
  AVX512VL256-NEXT: Cost Model: Found an estimated cost of 21 for instruction: %V32i8 = sext <32 x i1>
undef to <32 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 42 for instruction: %V64i8 = sext <64 x i1>
undef to <64 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; AVX512VL512-LABEL: 'sext256_vXi1'
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I64 = sext i1 undef to i64
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i64 = sext <2 x i1> undef
to <2 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i64 = sext <4 x i1> undef
to <4 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i64 = sext <8 x i1> undef
to <8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I32 = sext i1 undef to i32

```

```

; AVX512VL512-NEXT: Cost Model: Found an estimated
cost of 1 for instruction: %V2i32 = sext <2 x i1> undef to <2 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i32 = sext <4 x i1> undef
to <4 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i32 = sext <8 x i1> undef
to <8 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i32 = sext <16 x i1>
undef to <16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I16 = sext i1 undef to i16
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V2i16 = sext <2 x i1> undef
to <2 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V4i16 = sext <4 x i1> undef
to <4 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V8i16 = sext <8 x i1> undef
to <8 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V16i16 =
sext <16 x i1> undef to <16 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 7 for instruction: %V32i16 = sext <32 x i1>
undef to <32 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I8 = sext i1 undef to i8
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V2i8 = sext <2 x i1> undef to
<2 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V4i8 = sext <4 x i1> undef to
<4 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V8i8 = sext <8 x i1> undef to
<8 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V16i8 = sext <16 x i1> undef
to <16 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 7 for instruction: %V32i8 = sext <32 x i1> undef
to <32 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 15 for instruction: %V64i8 = sext <64 x i1>
undef to <64 x i8>
; AVX512VL512-NEXT:
Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; SKX256-LABEL: 'sext256_vXi1'
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I64 = sext i1 undef to i64
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i64 = sext <2 x i1> undef to <2 x
i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i64 = sext <4 x i1> undef to <4 x
i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i64 = sext <8 x i1> undef to <8 x
i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I32 = sext i1 undef to i32
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i32 = sext <2 x i1> undef to <2 x
i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i32 = sext <4 x i1> undef to <4 x
i32>

```

```

; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i32 = sext <8 x i1> undef
to <8 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i32 = sext <16 x i1> undef to
<16 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I16 = sext i1 undef to i16
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i16 = sext <2 x i1> undef to <2 x
i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i16 = sext <4 x i1> undef to <4 x
i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i16 = sext <8 x i1> undef to <8 x
i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i16 = sext <16 x i1> undef to
<16 x i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V32i16 = sext <32 x i1> undef to
<32 x i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I8 = sext i1 undef to i8
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i8
= sext <2 x i1> undef to <2 x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i8 = sext <4 x i1> undef to <4 x
i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i8 = sext <8 x i1> undef to <8 x
i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i8 = sext <16 x i1> undef to <16
x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V32i8 = sext <32 x i1> undef to <32
x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V64i8 = sext <64 x i1> undef to <64
x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; SKX512-LABEL: 'sext256_vXi1'
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I64 = sext i1 undef to i64
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i64 = sext <2 x i1> undef to <2 x
i64>
; SKX512-NEXT: Cost Model: Found an estimated
cost of 1 for instruction: %V4i64 = sext <4 x i1> undef to <4 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i64 = sext <8 x i1> undef to <8 x
i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I32 = sext i1 undef to i32
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i32 = sext <2 x i1> undef to <2 x
i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i32 = sext <4 x i1> undef to <4 x
i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i32 = sext <8 x i1> undef to <8 x
i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i32 = sext <16 x i1> undef to
<16 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I16 = sext i1 undef to i16

```



```

; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i16 = sext <2 x i1> undef to <2 x
i16>
; SKX512-NEXT: Cost Model:
Found an estimated cost of 1 for instruction: %V4i16 = sext <4 x i1> undef to <4 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i16 = sext <8 x i1> undef to <8 x
i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i16 = sext <16 x i1> undef to
<16 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V32i16 = sext <32 x i1> undef to
<32 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I8 = sext i1 undef to i8
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i8 = sext <2 x i1> undef to <2 x
i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i8 = sext <4 x i1> undef to <4 x
i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i8 = sext <8 x i1> undef to <8 x
i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i8 = sext <16 x i1> undef to <16
x i8>
;
SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V32i8 = sext <32 x i1> undef to <32
x i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V64i8 = sext <64 x i1> undef to <64
x i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
%I64 = sext i1 undef to i64
%V2i64 = sext <2 x i1> undef to <2 x i64>
%V4i64 = sext <4 x i1> undef to <4 x i64>
%V8i64 = sext <8 x i1> undef to <8 x i64>

%I32 = sext i1 undef to i32
%V2i32 = sext <2 x i1> undef to <2 x i32>
%V4i32 = sext <4 x i1> undef to <4 x i32>
%V8i32 = sext <8 x i1> undef to <8 x i32>
%V16i32 = sext <16 x i1> undef to <16 x i32>

%I16 = sext i1 undef to i16
%V2i16 = sext <2 x i1> undef to <2 x i16>
%V4i16 = sext <4 x i1> undef to <4 x i16>
%V8i16 = sext <8 x i1> undef to <8 x i16>
%V16i16 = sext <16 x i1> undef to <16 x i16>
%V32i16 = sext <32 x i1> undef to <32 x i16>

%I8 = sext i1 undef to i8
%V2i8 = sext <2
x i1> undef to <2 x i8>
%V4i8 = sext <4 x i1> undef to <4 x i8>

```

```

%V8i8 = sext <8 x i1> undef to <8 x i8>
%V16i8 = sext <16 x i1> undef to <16 x i8>
%V32i8 = sext <32 x i1> undef to <32 x i8>
%V64i8 = sext <64 x i1> undef to <64 x i8>

ret i32 undef
}

define i32 @trunc_vXi1() "min-legal-vector-width"="256" {
; AVX-LABEL: 'trunc_vXi1'
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V2i64 = trunc <2 x i64> undef to <2 x
i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V4i64 = trunc <4 x i64> undef to <4 x
i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 9 for instruction: %V8i64 = trunc <8 x i64> undef to <8 x
i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 11 for instruction: %V16i64 = trunc <16 x i64> undef to
<16 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i64 = trunc <32 x i64> undef to <32
x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V64i64
= trunc <64 x i64> undef to <64 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i32 = trunc <2 x i32> undef to <2 x
i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V4i32 = trunc <4 x i32> undef to <4 x
i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = trunc <8 x i32> undef to <8 x
i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 8 for instruction: %V16i32 = trunc <16 x i32> undef to <16
x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i32 = trunc <32 x i32> undef to <32
x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V64i32 = trunc <64 x i32> undef to <64
x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i16 = trunc <2 x i16> undef to <2 x
i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i16 = trunc <4 x i16> undef to <4 x
i1>
; AVX-NEXT: Cost Model: Found an estimated
cost of 0 for instruction: %V8i16 = trunc <8 x i16> undef to <8 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V16i16 = trunc <16 x i16> undef to <16
x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i16 = trunc <32 x i16> undef to <32
x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V64i16 = trunc <64 x i16> undef to <64
x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i8 = trunc <2 x i8> undef to <2 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i8 = trunc <4 x i8> undef to <4 x i1>

```

```

; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i8 = trunc <8 x i8> undef to <8 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V16i8 = trunc <16 x i8> undef to <16 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i8 = trunc <32 x i8> undef to <32 x i1>
; AVX-NEXT:
  Cost Model: Found an estimated cost of 0 for instruction: %V64i8 = trunc <64 x i8> undef to <64 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; AVX512VL256-LABEL: 'trunc_vXi1'
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = trunc <2 x i64> undef to <2 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = trunc <4 x i64> undef to <4 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 9 for instruction: %V8i64 = trunc <8 x i64> undef to <8 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 11 for instruction: %V16i64 = trunc <16 x i64> undef to <16 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i64 = trunc <32 x i64> undef to <32 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V64i64 = trunc <64 x i64> undef to <64 x i1>
; AVX512VL256-NEXT: Cost Model:
  Found an estimated cost of 2 for instruction: %V2i32 = trunc <2 x i32> undef to <2 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = trunc <4 x i32> undef to <4 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = trunc <8 x i32> undef to <8 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V16i32 = trunc <16 x i32> undef to <16 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 8 for instruction: %V32i32 = trunc <32 x i32> undef to <32 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 16 for instruction: %V64i32 = trunc <64 x i32> undef to <64 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V2i16 = trunc <2 x i16> undef to <2 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V4i16 = trunc <4 x i16> undef to <4 x i1>
; AVX512VL256-NEXT: Cost Model: Found an
  estimated cost of 3 for instruction: %V8i16 = trunc <8 x i16> undef to <8 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 8 for instruction: %V16i16 = trunc <16 x i16> undef to <16 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 16 for instruction: %V32i16 = trunc <32 x i16> undef to <32 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 32 for instruction: %V64i16 = trunc <64 x i16> undef to <64 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V2i8 = trunc <2 x i8> undef to <2 x i1>

```

```

; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V4i8 = trunc <4 x i8> undef
to <4 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V8i8 = trunc <8 x i8> undef
to <8 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 8 for instruction: %V16i8 = trunc <16 x i8>
undef to <16 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of
0 for instruction: %V32i8 = trunc <32 x i8> undef to <32 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V64i8 = trunc <64 x i8>
undef to <64 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; AVX512VL512-LABEL: 'trunc_vXi1'
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = trunc <2 x i64>
undef to <2 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = trunc <4 x i64>
undef to <4 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i64 = trunc <8 x i64>
undef to <8 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 11 for instruction: %V16i64 = trunc <16 x i64>
undef to <16 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i64 = trunc <32 x i64>
undef to <32 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction:
%V64i64 = trunc <64 x i64> undef to <64 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i32 = trunc <2 x i32>
undef to <2 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = trunc <4 x i32>
undef to <4 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = trunc <8 x i32>
undef to <8 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i32 = trunc <16 x i32>
undef to <16 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V32i32 = trunc <32 x i32>
undef to <32 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 8 for instruction: %V64i32 = trunc <64 x i32>
undef to <64 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V2i16 = trunc <2 x i16>
undef to <2 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction:
%V4i16 = trunc <4 x i16> undef to <4 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V8i16 = trunc <8 x i16>
undef to <8 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V16i16 = trunc <16 x i16>
undef to <16 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i16 = trunc <32 x i16>
undef to <32 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V64i16 = trunc <64 x i16>

```

```

undef to <64 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V2i8 = trunc <2 x i8> undef
to <2 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V4i8 = trunc <4 x i8> undef
to <4 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V8i8 = trunc <8 x i8> undef
to <8 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V16i8 = trunc <16 x
i8> undef to <16 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i8 = trunc <32 x i8>
undef to <32 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V64i8 = trunc <64 x i8>
undef to <64 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; SKX256-LABEL: 'trunc_vXi1'
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = trunc <2 x i64> undef to <2
x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = trunc <4 x i64> undef to <4
x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 9 for instruction: %V8i64 = trunc <8 x i64> undef to <8
x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 11 for instruction: %V16i64 = trunc <16 x i64> undef to
<16 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i64 = trunc <32 x i64> undef to
<32 x i1>
; SKX256-NEXT:
Cost Model: Found an estimated cost of 0 for instruction: %V64i64 = trunc <64 x i64> undef to <64 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i32 = trunc <2 x i32> undef to <2
x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = trunc <4 x i32> undef to <4
x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = trunc <8 x i32> undef to <8
x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V16i32 = trunc <16 x i32> undef to
<16 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i32 = trunc <32 x i32> undef to
<32 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V64i32 = trunc <64 x i32> undef to
<64 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i16 = trunc <2 x i16> undef to <2
x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction:
%V4i16 = trunc <4 x i16> undef to <4 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i16 = trunc <8 x i16> undef to <8
x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i16 = trunc <16 x i16> undef to
<16 x i1>

```

```

; SKX256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V32i16 = trunc <32 x i16> undef to
<32 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 8 for instruction: %V64i16 = trunc <64 x i16> undef to
<64 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i8 = trunc <2 x i8> undef to <2 x
i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i8 = trunc <4 x i8> undef to <4 x
i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i8 = trunc <8 x i8> undef to <8 x
i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i8 = trunc <16 x i8> undef to
<16 x i1>
; SKX256-NEXT:
Cost Model: Found an estimated cost of 2 for instruction: %V32i8 = trunc <32 x i8> undef to <32 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V64i8 = trunc <64 x i8> undef to
<64 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; SKX512-LABEL: 'trunc_vXi1'
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = trunc <2 x i64> undef to <2
x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = trunc <4 x i64> undef to <4
x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i64 = trunc <8 x i64> undef to <8
x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 11 for instruction: %V16i64 = trunc <16 x i64> undef to
<16 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i64 = trunc <32 x i64> undef to
<32 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction:
%V64i64 = trunc <64 x i64> undef to <64 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i32 = trunc <2 x i32> undef to <2
x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = trunc <4 x i32> undef to <4
x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = trunc <8 x i32> undef to <8
x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i32 = trunc <16 x i32> undef to
<16 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i32 = trunc <32 x i32> undef to
<32 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V64i32 = trunc <64 x i32> undef to
<64 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i16 = trunc <2 x i16> undef to <2
x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i16 = trunc <4 x i16> undef to <4
x i1>
;

```

SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i16 = trunc <8 x i16> undef to <8 x i1>

; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i16 = trunc <16 x i16> undef to <16 x i1>

; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V32i16 = trunc <32 x i16> undef to <32 x i1>

; SKX512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V64i16 = trunc <64 x i16> undef to <64 x i1>

; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i8 = trunc <2 x i8> undef to <2 x i1>

; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i8 = trunc <4 x i8> undef to <4 x i1>

; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i8 = trunc <8 x i8> undef to <8 x i1>

; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i8 = trunc <16 x i8> undef to <16 x i1>

; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V32i8 = trunc <32 x i8> undef to <32 x i1>

; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V64i8 = trunc <64 x i8> undef to <64 x i1>

; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef

;

%V2i64 = trunc <2 x i64> undef to <2 x i1>

%V4i64 = trunc <4 x i64> undef to <4 x i1>

%V8i64 = trunc <8 x i64> undef to <8 x i1>

%V16i64 = trunc <16 x i64> undef to <16 x i1>

%V32i64 = trunc <32 x i64> undef to <32 x i1>

%V64i64 = trunc <64 x i64> undef to <64 x i1>

%V2i32 = trunc <2 x i32> undef to <2 x i1>

%V4i32 = trunc <4 x i32> undef to <4 x i1>

%V8i32 = trunc <8 x i32> undef to <8 x i1>

%V16i32 = trunc <16 x i32> undef to <16 x i1>

%V32i32 = trunc <32 x i32> undef to <32 x i1>

%V64i32 = trunc <64 x i32> undef to <64 x i1>

%V2i16 = trunc <2 x i16> undef to <2 x i1>

%V4i16 = trunc <4 x i16> undef to <4 x i1>

%V8i16 = trunc <8 x i16> undef to <8 x i1>

%V16i16 = trunc <16 x i16> undef to <16 x i1>

%V32i16 = trunc <32 x i16> undef to <32 x i1>

%V64i16 = trunc <64 x i16> undef to <64 x i1>

%V2i8 = trunc <2 x i8> undef to <2 x i1>

%V4i8 = trunc <4 x i8> undef to <4 x i1>

%V8i8 = trunc <8 x i8> undef to <8 x i1>

%V16i8 = trunc <16 x i8> undef to <16 x i1>

%V32i8 = trunc <32 x i8> undef to <32 x i1>

%V64i8 = trunc <64 x i8> undef to <64 x i1>

ret i32 undef

}

=====  
The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:  
=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached



to the work

(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition,

"submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason

of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

### ---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

=====

Software from third parties included in the LLVM Project:

=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or `LICENSE` file at the top containing the specific license and restrictions which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every file.

=====

Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):

=====

The libc++abi library is dual licensed under both the University of Illinois "BSD-Like" license and the MIT license. As a user of this code you may choose to use it under either license. As a contributor, you agree to allow your code to be used under both.

Full text of the relevant licenses is included below.

=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2009-2019 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

<http://llvm.org>

Permission

is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- \* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====

Copyright (c) 2009-2014 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

; NOTE: Assertions have been autogenerated by utils/update\_llc\_test\_checks.py

; RUN: llc -mtriple=riscv64 -mattr=+m -verify-machineinstrs < %s \

; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64

;

```

; Get the actual value of the overflow bit.
;
define zeroext i1 @saddo1.i32(i32 signext %v1, i32 signext %v2, ptr %res) {
; RV64-LABEL: saddo1.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addw a3, a0, a1
; RV64-NEXT:  slt a0, a3, a0
; RV64-NEXT:  slti a1, a1, 0
; RV64-NEXT:  xor a0, a1, a0
; RV64-NEXT:  sw a3, 0(a2)
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.sadd.with.overflow.i32(i32 %v1, i32 %v2)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

```

```

; Test the immediate version.
define zeroext i1 @saddo2.i32(i32 signext %v1, ptr %res) {
; RV64-LABEL: saddo2.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addiw a2, a0, 4
; RV64-NEXT:  slt a0, a2, a0
; RV64-NEXT:
; RV64-NEXT:  sw a2, 0(a1)
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.sadd.with.overflow.i32(i32 %v1, i32 4)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

```

```

; Test negative immediates.
define zeroext i1 @saddo3.i32(i32 signext %v1, ptr %res) {
; RV64-LABEL: saddo3.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addiw a2, a0, -4
; RV64-NEXT:  slt a0, a2, a0
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  sw a2, 0(a1)
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.sadd.with.overflow.i32(i32 %v1, i32 -4)
%val = extractvalue {i32, i1} %t, 0

```

```

%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

; Test immediates that are too large to be encoded.
define zeroext i1 @saddo4.i32(i32 signext %v1, ptr %res) {
; RV64-LABEL: saddo4.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT: lui a2, 4096
; RV64-NEXT: addi a2, a2, -1
; RV64-NEXT: addw a2, a0, a2
; RV64-NEXT: slt
a0, a2, a0
; RV64-NEXT: sw a2, 0(a1)
; RV64-NEXT: ret
entry:
%t = call {i32, i1} @llvm.sadd.with.overflow.i32(i32 %v1, i32 16777215)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @saddo1.i64(i64 %v1, i64 %v2, ptr %res) {
; RV64-LABEL: saddo1.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT: add a3, a0, a1
; RV64-NEXT: slt a0, a3, a0
; RV64-NEXT: slti a1, a1, 0
; RV64-NEXT: xor a0, a1, a0
; RV64-NEXT: sd a3, 0(a2)
; RV64-NEXT: ret
entry:
%t = call {i64, i1} @llvm.sadd.with.overflow.i64(i64 %v1, i64 %v2)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
store i64 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @saddo2.i64(i64 %v1, ptr %res) {
; RV64-LABEL: saddo2.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT: addi a2, a0, 4
; RV64-NEXT: slt a0, a2, a0
; RV64-NEXT: sd a2, 0(a1)
; RV64-NEXT: ret

```



```

entry:
    %t = call {i64, i1}
    @llvm.sadd.with.overflow.i64(i64 %v1, i64 4)
    %val = extractvalue {i64, i1} %t, 0
    %obit = extractvalue {i64, i1} %t, 1
    store i64 %val, ptr %res
    ret i1 %obit
}

define zeroext i1 @saddo3.i64(i64 %v1, ptr %res) {
; RV64-LABEL: saddo3.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addi a2, a0, -4
; RV64-NEXT:  slt a0, a2, a0
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  sd a2, 0(a1)
; RV64-NEXT:  ret
entry:
    %t = call {i64, i1} @llvm.sadd.with.overflow.i64(i64 %v1, i64 -4)
    %val = extractvalue {i64, i1} %t, 0
    %obit = extractvalue {i64, i1} %t, 1
    store i64 %val, ptr %res
    ret i1 %obit
}

define zeroext i1 @uaddo.i32(i32 signext %v1, i32 signext %v2, ptr %res) {
; RV64-LABEL: uaddo.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addw a1, a0, a1
; RV64-NEXT:  sltu a0, a1, a0
; RV64-NEXT:  sw a1, 0(a2)
; RV64-NEXT:  ret
entry:
    %t = call {i32, i1} @llvm.uadd.with.overflow.i32(i32 %v1, i32 %v2)
    %val = extractvalue {i32, i1} %t, 0
    %obit = extractvalue {i32,
i1} %t, 1
    store i32 %val, ptr %res
    ret i1 %obit
}

define zeroext i1 @uaddo.i32.constant(i32 signext %v1, ptr %res) {
; RV64-LABEL: uaddo.i32.constant:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addiw a2, a0, -2
; RV64-NEXT:  sltu a0, a2, a0
; RV64-NEXT:  sw a2, 0(a1)
; RV64-NEXT:  ret

```

```

entry:
    %t = call {i32, i1} @llvm.uadd.with.overflow.i32(i32 %v1, i32 -2)
    %val = extractvalue {i32, i1} %t, 0
    %obit = extractvalue {i32, i1} %t, 1
    store i32 %val, ptr %res
    ret i1 %obit
}

define zeroext i1 @uaddo.i32.constant_one(i32 signext %v1, ptr %res) {
; RV64-LABEL: uaddo.i32.constant_one:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addiw a2, a0, 1
; RV64-NEXT:  seqz a0, a2
; RV64-NEXT:  sw a2, 0(a1)
; RV64-NEXT:  ret
entry:
    %t = call {i32, i1} @llvm.uadd.with.overflow.i32(i32 %v1, i32 1)
    %val = extractvalue {i32, i1} %t, 0
    %obit = extractvalue {i32, i1} %t, 1
    store i32 %val, ptr %res
    ret i1 %obit
}

define zeroext i1 @uaddo.i64(i64 %v1, i64 %v2, ptr
    %res) {
; RV64-LABEL: uaddo.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  add a1, a0, a1
; RV64-NEXT:  sltu a0, a1, a0
; RV64-NEXT:  sd a1, 0(a2)
; RV64-NEXT:  ret
entry:
    %t = call {i64, i1} @llvm.uadd.with.overflow.i64(i64 %v1, i64 %v2)
    %val = extractvalue {i64, i1} %t, 0
    %obit = extractvalue {i64, i1} %t, 1
    store i64 %val, ptr %res
    ret i1 %obit
}

define zeroext i1 @uaddo.i64.constant_one(i64 %v1, ptr %res) {
; RV64-LABEL: uaddo.i64.constant_one:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addi a2, a0, 1
; RV64-NEXT:  seqz a0, a2
; RV64-NEXT:  sd a2, 0(a1)
; RV64-NEXT:  ret
entry:
    %t = call {i64, i1} @llvm.uadd.with.overflow.i64(i64 %v1, i64 1)

```

```

%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
store i64 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @ssubo1.i32(i32 signext %v1, i32 signext %v2, ptr %res) {
; RV64-LABEL: ssubo1.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  sgtz a3, a1
; RV64-NEXT:  subw a1,
a0, a1
; RV64-NEXT:  slt a0, a1, a0
; RV64-NEXT:  xor a0, a3, a0
; RV64-NEXT:  sw a1, 0(a2)
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.ssub.with.overflow.i32(i32 %v1, i32 %v2)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @ssubo2.i32(i32 signext %v1, ptr %res) {
; RV64-LABEL: ssubo2.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addiw a2, a0, 4
; RV64-NEXT:  slt a0, a2, a0
; RV64-NEXT:  sw a2, 0(a1)
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.ssub.with.overflow.i32(i32 %v1, i32 -4)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @ssubo.i64(i64 %v1, i64 %v2, ptr %res) {
; RV64-LABEL: ssubo.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  sgtz a3, a1
; RV64-NEXT:  sub a1, a0, a1
; RV64-NEXT:  slt a0, a1, a0
; RV64-NEXT:  xor a0, a3, a0
; RV64-NEXT:  sd a1,

```

```

0(a2)
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.ssub.with.overflow.i64(i64 %v1, i64 %v2)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
store i64 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @usubo.i32(i32 signext %v1, i32 signext %v2, ptr %res) {
; RV64-LABEL: usubo.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  subw a1, a0, a1
; RV64-NEXT:  sltu a0, a0, a1
; RV64-NEXT:  sw a1, 0(a2)
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.usub.with.overflow.i32(i32 %v1, i32 %v2)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @usubo.i32.constant.rhs(i32 signext %v1, ptr %res) {
; RV64-LABEL: usubo.i32.constant.rhs:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addiw a2, a0, 2
; RV64-NEXT:  sltu a0, a0, a2
; RV64-NEXT:  sw a2, 0(a1)
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.usub.with.overflow.i32(i32 %v1, i32 -2)

%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @usubo.i32.constant.lhs(i32 signext %v1, ptr %res) {
; RV64-LABEL: usubo.i32.constant.lhs:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  li a2, -2
; RV64-NEXT:  subw a2, a2, a0
; RV64-NEXT:  addi a0, a2, 1
; RV64-NEXT:  seqz a0, a0

```

```

; RV64-NEXT:  sw a2, 0(a1)
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.usub.with.overflow.i32(i32 -2, i32 %v1)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @usubo.i64(i64 %v1, i64 %v2, ptr %res) {
; RV64-LABEL: usubo.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  sub a1, a0, a1
; RV64-NEXT:  sltu a0, a0, a1
; RV64-NEXT:  sd a1, 0(a2)
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.usub.with.overflow.i64(i64 %v1, i64 %v2)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t,
1
store i64 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @smulo.i32(i32 signext %v1, i32 signext %v2, ptr %res) {
; RV64-LABEL: smulo.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  mul a1, a0, a1
; RV64-NEXT:  srai a0, a1, 32
; RV64-NEXT:  sraiw a3, a1, 31
; RV64-NEXT:  xor a0, a0, a3
; RV64-NEXT:  snez a0, a0
; RV64-NEXT:  sw a1, 0(a2)
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.smul.with.overflow.i32(i32 %v1, i32 %v2)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @smulo2.i32(i32 signext %v1, ptr %res) {
; RV64-LABEL: smulo2.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  li a2, 13

```

```

; RV64-NEXT:  mul a2, a0, a2
; RV64-NEXT:  srai a0, a2, 32
; RV64-NEXT:  sraiw a3, a2, 31
; RV64-NEXT:  xor a0, a0, a3
; RV64-NEXT:  snez a0, a0
; RV64-NEXT:  sw a2, 0(a1)
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.smul.with.overflow.i32(i32 %v1, i32
13)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @smulo.i64(i64 %v1, i64 %v2, ptr %res) {
; RV64-LABEL: smulo.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  mulh a3, a0, a1
; RV64-NEXT:  mul a1, a0, a1
; RV64-NEXT:  srai a0, a1, 63
; RV64-NEXT:  xor a0, a3, a0
; RV64-NEXT:  snez a0, a0
; RV64-NEXT:  sd a1, 0(a2)
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.smul.with.overflow.i64(i64 %v1, i64 %v2)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
store i64 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @smulo2.i64(i64 %v1, ptr %res) {
; RV64-LABEL: smulo2.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  li a2, 13
; RV64-NEXT:  mulh a3, a0, a2
; RV64-NEXT:  mul a2, a0, a2
; RV64-NEXT:  srai a0, a2, 63
; RV64-NEXT:  xor a0, a3, a0
; RV64-NEXT:  snez a0, a0
; RV64-NEXT:  sd a2, 0(a1)
; RV64-NEXT:  ret
entry:
%t = call

```

```

{i64, i1} @llvm.smul.with.overflow.i64(i64 %v1, i64 13)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
store i64 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @umulo.i32(i32 signext %v1, i32 signext %v2, ptr %res) {
; RV64-LABEL: umulo.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  slli a1, a1, 32
; RV64-NEXT:  slli a0, a0, 32
; RV64-NEXT:  mulhu a1, a0, a1
; RV64-NEXT:  srli a0, a1, 32
; RV64-NEXT:  snez a0, a0
; RV64-NEXT:  sw a1, 0(a2)
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.umul.with.overflow.i32(i32 %v1, i32 %v2)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

define zeroext i1 @umulo2.i32(i32 signext %v1, ptr %res) {
; RV64-LABEL: umulo2.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  li a2, 13
; RV64-NEXT:  slli a2, a2, 32
; RV64-NEXT:  slli a0, a0, 32
; RV64-NEXT:  mulhu a2, a0, a2
; RV64-NEXT:  srli a0, a2, 32
; RV64-NEXT:
; RV64-NEXT:  snez a0, a0
; RV64-NEXT:  sw a2, 0(a1)
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.umul.with.overflow.i32(i32 %v1, i32 13)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
store i32 %val, ptr %res
ret i1 %obit
}

; Similar to umulo.i32, but storing the overflow and returning the result.
define signext i32 @umulo3.i32(i32 signext %0, i32 signext %1, ptr %2) {
; RV64-LABEL: umulo3.i32:

```

```

; RV64:    # %bb.0:
; RV64-NEXT:  slli a1, a1, 32
; RV64-NEXT:  slli a0, a0, 32
; RV64-NEXT:  mulhu a0, a0, a1
; RV64-NEXT:  srai a1, a0, 32
; RV64-NEXT:  snez a1, a1
; RV64-NEXT:  sext.w a0, a0
; RV64-NEXT:  sw a1, 0(a2)
; RV64-NEXT:  ret
%4 = tail call { i32, i1 } @llvm.umul.with.overflow.i32(i32 %0, i32 %1)
%5 = extractvalue { i32, i1 } %4, 1
%6 = extractvalue { i32, i1 } %4, 0
%7 = zext i1 %5 to i32
store i32 %7, ptr %2, align 4
ret i32 %6
}

```

```

define zeroext i1 @umulo.i64(i64 %v1, i64 %v2, ptr %res) {
; RV64-LABEL: umulo.i64:
;
; RV64:    # %bb.0: # %entry
; RV64-NEXT:  mulhu a3, a0, a1
; RV64-NEXT:  snez a3, a3
; RV64-NEXT:  mul a0, a0, a1
; RV64-NEXT:  sd a0, 0(a2)
; RV64-NEXT:  mv a0, a3
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.umul.with.overflow.i64(i64 %v1, i64 %v2)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
store i64 %val, ptr %res
ret i1 %obit
}

```

```

define zeroext i1 @umulo2.i64(i64 %v1, ptr %res) {
; RV64-LABEL: umulo2.i64:
; RV64:    # %bb.0: # %entry
; RV64-NEXT:  li a3, 13
; RV64-NEXT:  mulhu a2, a0, a3
; RV64-NEXT:  snez a2, a2
; RV64-NEXT:  mul a0, a0, a3
; RV64-NEXT:  sd a0, 0(a1)
; RV64-NEXT:  mv a0, a2
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.umul.with.overflow.i64(i64 %v1, i64 13)
%val = extractvalue {i64, i1} %t, 0

```



```

%obit = extractvalue {i64, i1} %t, 1
store i64 %val, ptr %res
ret i1 %obit
}

;
; Check the use of the overflow bit in combination with a select instruction.
;
define i32 @saddo.select.i32(i32
  signext %v1, i32 signext %v2) {
; RV64-LABEL: saddo.select.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addw a2, a0, a1
; RV64-NEXT:  slt a2, a2, a0
; RV64-NEXT:  slti a3, a1, 0
; RV64-NEXT:  bne a3, a2, .LBB28_2
; RV64-NEXT:  # %bb.1: # %entry
; RV64-NEXT:  mv a0, a1
; RV64-NEXT:  .LBB28_2: # %entry
; RV64-NEXT:  ret
entry:
  %t = call {i32, i1} @llvm.sadd.with.overflow.i32(i32 %v1, i32 %v2)
  %obit = extractvalue {i32, i1} %t, 1
  %ret = select i1 %obit, i32 %v1, i32 %v2
  ret i32 %ret
}

define i1 @saddo.not.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL: saddo.not.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addw a2, a0, a1
; RV64-NEXT:  slt a0, a2, a0
; RV64-NEXT:  slti a1, a1, 0
; RV64-NEXT:  xor a0, a1, a0
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  ret
entry:
  %t = call {i32, i1} @llvm.sadd.with.overflow.i32(i32 %v1, i32 %v2)
  %obit = extractvalue {i32, i1} %t, 1
  %ret = xor i1 %obit, true
  ret i1 %ret
}

define i64 @saddo.select.i64(i64
  %v1, i64 %v2) {
; RV64-LABEL: saddo.select.i64:
; RV64:      # %bb.0: # %entry

```

```

; RV64-NEXT:  add a2, a0, a1
; RV64-NEXT:  slt a2, a2, a0
; RV64-NEXT:  slti a3, a1, 0
; RV64-NEXT:  bne a3, a2, .LBB30_2
; RV64-NEXT:  # %bb.1: # %entry
; RV64-NEXT:  mv a0, a1
; RV64-NEXT:  .LBB30_2: # %entry
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.sadd.with.overflow.i64(i64 %v1, i64 %v2)
%obit = extractvalue {i64, i1} %t, 1
%ret = select i1 %obit, i64 %v1, i64 %v2
ret i64 %ret
}

define i1 @saddo.not.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: saddo.not.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  add a2, a0, a1
; RV64-NEXT:  slt a0, a2, a0
; RV64-NEXT:  slti a1, a1, 0
; RV64-NEXT:  xor a0, a1, a0
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.sadd.with.overflow.i64(i64 %v1, i64 %v2)
%obit = extractvalue {i64, i1} %t, 1
%ret = xor i1 %obit, true
ret i1 %ret
}

define i32 @uaddo.select.i32(i32 signext %v1,
i32 signext %v2) {
; RV64-LABEL: uaddo.select.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addw a2, a0, a1
; RV64-NEXT:  bltu a2, a0, .LBB32_2
; RV64-NEXT:  # %bb.1: # %entry
; RV64-NEXT:  mv a0, a1
; RV64-NEXT:  .LBB32_2: # %entry
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.uadd.with.overflow.i32(i32 %v1, i32 %v2)
%obit = extractvalue {i32, i1} %t, 1
%ret = select i1 %obit, i32 %v1, i32 %v2
ret i32 %ret
}

```

```

define i1 @uaddo.not.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL: uaddo.not.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addw a1, a0, a1
; RV64-NEXT:  sltu a0, a1, a0
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.uadd.with.overflow.i32(i32 %v1, i32 %v2)
%obit = extractvalue {i32, i1} %t, 1
%ret = xor i1 %obit, true
ret i1 %ret
}

```

```

define i64 @uaddo.select.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: uaddo.select.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  add a2, a0, a1
; RV64-NEXT:
; RV64-NEXT:  bltu a2, a0, .LBB34_2
; RV64-NEXT:  # %bb.1: # %entry
; RV64-NEXT:  mv a0, a1
; RV64-NEXT:  .LBB34_2: # %entry
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.uadd.with.overflow.i64(i64 %v1, i64 %v2)
%obit = extractvalue {i64, i1} %t, 1
%ret = select i1 %obit, i64 %v1, i64 %v2
ret i64 %ret
}

```

```

define i1 @uaddo.not.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: uaddo.not.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  add a1, a0, a1
; RV64-NEXT:  sltu a0, a1, a0
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.uadd.with.overflow.i64(i64 %v1, i64 %v2)
%obit = extractvalue {i64, i1} %t, 1
%ret = xor i1 %obit, true
ret i1 %ret
}

```

```

define i32 @ssubo.select.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL: ssubo.select.i32:
; RV64:      # %bb.0: # %entry

```

```

; RV64-NEXT:  sgtz a2, a1
; RV64-NEXT:  subw a3, a0, a1
; RV64-NEXT:  slt a3, a3, a0
; RV64-NEXT:  bne a2, a3, .LBB36_2
; RV64-NEXT:  # %bb.1: # %entry
; RV64-NEXT:  mv a0,
a1
; RV64-NEXT:  .LBB36_2: # %entry
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.ssub.with.overflow.i32(i32 %v1, i32 %v2)
%obit = extractvalue {i32, i1} %t, 1
%ret = select i1 %obit, i32 %v1, i32 %v2
ret i32 %ret
}

```

```

define i1 @ssubo.not.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL: ssubo.not.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  sgtz a2, a1
; RV64-NEXT:  subw a1, a0, a1
; RV64-NEXT:  slt a0, a1, a0
; RV64-NEXT:  xor a0, a2, a0
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.ssub.with.overflow.i32(i32 %v1, i32 %v2)
%obit = extractvalue {i32, i1} %t, 1
%ret = xor i1 %obit, true
ret i1 %ret
}

```

```

define i64 @ssubo.select.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: ssubo.select.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  sgtz a2, a1
; RV64-NEXT:  sub a3, a0, a1
; RV64-NEXT:  slt a3, a3, a0
; RV64-NEXT:  bne a2, a3, .LBB38_2
; RV64-NEXT:  # %bb.1: # %entry
; RV64-NEXT:  mv a0, a1
; RV64-NEXT:
.LBB38_2: # %entry
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.ssub.with.overflow.i64(i64 %v1, i64 %v2)
%obit = extractvalue {i64, i1} %t, 1
%ret = select i1 %obit, i64 %v1, i64 %v2

```

```

ret i64 %ret
}

define i1 @ssub.not.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: ssub.not.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  sgtz a2, a1
; RV64-NEXT:  sub a1, a0, a1
; RV64-NEXT:  slt a0, a1, a0
; RV64-NEXT:  xor a0, a2, a0
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.ssub.with.overflow.i64(i64 %v1, i64 %v2)
%obit = extractvalue {i64, i1} %t, 1
%ret = xor i1 %obit, true
ret i1 %ret
}

define i32 @usubo.select.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL: usubo.select.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  subw a2, a0, a1
; RV64-NEXT:  bltu a0, a2, .LBB40_2
; RV64-NEXT:  # %bb.1: # %entry
; RV64-NEXT:  mv a0, a1
; RV64-NEXT:  .LBB40_2: # %entry
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.usub.with.overflow.i32(i32
%v1, i32 %v2)
%obit = extractvalue {i32, i1} %t, 1
%ret = select i1 %obit, i32 %v1, i32 %v2
ret i32 %ret
}

define i1 @usubo.not.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL: usubo.not.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  subw a1, a0, a1
; RV64-NEXT:  sltu a0, a0, a1
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.usub.with.overflow.i32(i32 %v1, i32 %v2)
%obit = extractvalue {i32, i1} %t, 1
%ret = xor i1 %obit, true
ret i1 %ret
}

```

```

}

define i64 @usubo.select.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: usubo.select.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  sub a2, a0, a1
; RV64-NEXT:  bltu a0, a2, .LBB42_2
; RV64-NEXT:  # %bb.1: # %entry
; RV64-NEXT:  mv a0, a1
; RV64-NEXT:  .LBB42_2: # %entry
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.usub.with.overflow.i64(i64 %v1, i64 %v2)
%obit = extractvalue {i64, i1} %t, 1
%ret = select i1 %obit, i64 %v1, i64 %v2
ret i64 %ret
}

```

```

define
i1 @usubo.not.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: usubo.not.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  sub a1, a0, a1
; RV64-NEXT:  sltu a0, a0, a1
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.usub.with.overflow.i64(i64 %v1, i64 %v2)
%obit = extractvalue {i64, i1} %t, 1
%ret = xor i1 %obit, true
ret i1 %ret
}

```

```

define i32 @smulo.select.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL: smulo.select.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  mul a2, a0, a1
; RV64-NEXT:  srai a3, a2, 32
; RV64-NEXT:  sraiw a2, a2, 31
; RV64-NEXT:  bne a3, a2, .LBB44_2
; RV64-NEXT:  # %bb.1: # %entry
; RV64-NEXT:  mv a0, a1
; RV64-NEXT:  .LBB44_2: # %entry
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.smul.with.overflow.i32(i32 %v1, i32 %v2)
%obit = extractvalue {i32, i1} %t, 1
%ret = select i1 %obit, i32 %v1, i32 %v2

```

```

ret i32 %ret
}

define i1 @smulo.not.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL:
smulo.not.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  mul a0, a0, a1
; RV64-NEXT:  srai a1, a0, 32
; RV64-NEXT:  sraiw a0, a0, 31
; RV64-NEXT:  xor a0, a1, a0
; RV64-NEXT:  snez a0, a0
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.smul.with.overflow.i32(i32 %v1, i32 %v2)
%obit = extractvalue {i32, i1} %t, 1
%ret = xor i1 %obit, true
ret i1 %ret
}

define i64 @smulo.select.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: smulo.select.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  mulh a2, a0, a1
; RV64-NEXT:  mul a3, a0, a1
; RV64-NEXT:  srai a3, a3, 63
; RV64-NEXT:  bne a2, a3, .LBB46_2
; RV64-NEXT:  # %bb.1: # %entry
; RV64-NEXT:  mv a0, a1
; RV64-NEXT:  .LBB46_2: # %entry
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.smul.with.overflow.i64(i64 %v1, i64 %v2)
%obit = extractvalue {i64, i1} %t, 1
%ret = select i1 %obit, i64 %v1, i64 %v2
ret i64 %ret
}

define i1 @smulo.not.i64(i64 %v1, i64 %v2) {
; RV64-LABEL:
smulo.not.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  mulh a2, a0, a1
; RV64-NEXT:  mul a0, a0, a1
; RV64-NEXT:  srai a0, a0, 63
; RV64-NEXT:  xor a0, a2, a0
; RV64-NEXT:  snez a0, a0

```

```

; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.smul.with.overflow.i64(i64 %v1, i64 %v2)
%obit = extractvalue {i64, i1} %t, 1
%ret = xor i1 %obit, true
ret i1 %ret
}

```

```

define i32 @umulo.select.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL: umulo.select.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  slli a2, a1, 32
; RV64-NEXT:  slli a3, a0, 32
; RV64-NEXT:  mulhu a2, a3, a2
; RV64-NEXT:  srai a2, a2, 32
; RV64-NEXT:  bnez a2, .LBB48_2
; RV64-NEXT: # %bb.1: # %entry
; RV64-NEXT:  mv a0, a1
; RV64-NEXT: .LBB48_2: # %entry
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.umul.with.overflow.i32(i32 %v1, i32 %v2)
%obit = extractvalue {i32, i1} %t, 1
%ret = select i1 %obit, i32 %v1, i32 %v2
ret i32 %ret
}

```

```

define
i1 @umulo.not.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL: umulo.not.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  slli a1, a1, 32
; RV64-NEXT:  slli a0, a0, 32
; RV64-NEXT:  mulhu a0, a0, a1
; RV64-NEXT:  srai a0, a0, 32
; RV64-NEXT:  snez a0, a0
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.umul.with.overflow.i32(i32 %v1, i32 %v2)
%obit = extractvalue {i32, i1} %t, 1
%ret = xor i1 %obit, true
ret i1 %ret
}

```

```

define i64 @umulo.select.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: umulo.select.i64:

```



```

; RV64:    # %bb.0: # %entry
; RV64-NEXT:  mulhu a2, a0, a1
; RV64-NEXT:  bnez a2, .LBB50_2
; RV64-NEXT:  # %bb.1: # %entry
; RV64-NEXT:  mv a0, a1
; RV64-NEXT:  .LBB50_2: # %entry
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.umul.with.overflow.i64(i64 %v1, i64 %v2)
%obit = extractvalue {i64, i1} %t, 1
%ret = select i1 %obit, i64 %v1, i64 %v2
ret i64 %ret
}

```

```

define i1 @umulo.not.i64(i64 %v1, i64 %v2)
{
; RV64-LABEL: umulo.not.i64:
; RV64:    # %bb.0: # %entry
; RV64-NEXT:  mulhu a0, a0, a1
; RV64-NEXT:  snez a0, a0
; RV64-NEXT:  xori a0, a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.umul.with.overflow.i64(i64 %v1, i64 %v2)
%obit = extractvalue {i64, i1} %t, 1
%ret = xor i1 %obit, true
ret i1 %ret
}

```

```

;
; Check the use of the overflow bit in combination with a branch instruction.
;

```

```

define zeroext i1 @saddo.br.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL: saddo.br.i32:
; RV64:    # %bb.0: # %entry
; RV64-NEXT:  addw a2, a0, a1
; RV64-NEXT:  slt a0, a2, a0
; RV64-NEXT:  slti a1, a1, 0
; RV64-NEXT:  beq a1, a0, .LBB52_2
; RV64-NEXT:  # %bb.1: # %overflow
; RV64-NEXT:  li a0, 0
; RV64-NEXT:  ret
; RV64-NEXT:  .LBB52_2: # %continue
; RV64-NEXT:  li a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.sadd.with.overflow.i32(i32 %v1, i32 %v2)

```

```

%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
br i1
%obit, label %overflow, label %continue

overflow:
ret i1 false

continue:
ret i1 true
}

define zeroext i1 @saddo.br.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: saddo.br.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  add a2, a0, a1
; RV64-NEXT:  slt a0, a2, a0
; RV64-NEXT:  slti a1, a1, 0
; RV64-NEXT:  beq a1, a0, .LBB53_2
; RV64-NEXT:  # %bb.1: # %overflow
; RV64-NEXT:  li a0, 0
; RV64-NEXT:  ret
; RV64-NEXT: .LBB53_2: # %continue
; RV64-NEXT:  li a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.sadd.with.overflow.i64(i64 %v1, i64 %v2)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
br i1 %obit, label %overflow, label %continue

overflow:
ret i1 false

continue:
ret i1 true
}

define zeroext i1 @uaddo.br.i32(i32 %v1, i32 %v2) {
; RV64-LABEL: uaddo.br.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addw a1, a0, a1
; RV64-NEXT:  sext.w a0, a0
; RV64-NEXT:  bgeu a1, a0, .LBB54_2
; RV64-NEXT:  # %bb.1: # %overflow
;
; RV64-NEXT:  li a0, 0
; RV64-NEXT:  ret

```

```

; RV64-NEXT: .LBB54_2: # %continue
; RV64-NEXT:  li a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.uadd.with.overflow.i32(i32 %v1, i32 %v2)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
br i1 %obit, label %overflow, label %continue

overflow:
ret i1 false

continue:
ret i1 true
}

```

```

define zeroext i1 @uaddo.br.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: uaddo.br.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  add a1, a0, a1
; RV64-NEXT:  bgeu a1, a0, .LBB55_2
; RV64-NEXT:  # %bb.1: # %overflow
; RV64-NEXT:  li a0, 0
; RV64-NEXT:  ret
; RV64-NEXT: .LBB55_2: # %continue
; RV64-NEXT:  li a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.uadd.with.overflow.i64(i64 %v1, i64 %v2)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
br i1 %obit, label %overflow, label %continue

overflow:
ret i1 false

continue:
ret i1 true
}

```

```

define zeroext i1 @ssubo.br.i32(i32
signext %v1, i32 signext %v2) {
; RV64-LABEL: ssubo.br.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  sgtz a2, a1
; RV64-NEXT:  subw a1, a0, a1
; RV64-NEXT:  slt a0, a1, a0
; RV64-NEXT:  beq a2, a0, .LBB56_2

```

```

; RV64-NEXT: # %bb.1: # %overflow
; RV64-NEXT: li a0, 0
; RV64-NEXT: ret
; RV64-NEXT: .LBB56_2: # %continue
; RV64-NEXT: li a0, 1
; RV64-NEXT: ret
entry:
%t = call {i32, i1} @llvm.ssub.with.overflow.i32(i32 %v1, i32 %v2)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
br i1 %obit, label %overflow, label %continue

overflow:
ret i1 false

continue:
ret i1 true
}

```

```

define zeroext i1 @ssubo.br.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: ssubo.br.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT: sgtz a2, a1
; RV64-NEXT: sub a1, a0, a1
; RV64-NEXT: slt a0, a1, a0
; RV64-NEXT: beq a2, a0, .LBB57_2
; RV64-NEXT: # %bb.1: # %overflow
; RV64-NEXT: li a0, 0
; RV64-NEXT: ret
; RV64-NEXT: .LBB57_2: # %continue
;
RV64-NEXT: li a0, 1
; RV64-NEXT: ret
entry:
%t = call {i64, i1} @llvm.ssub.with.overflow.i64(i64 %v1, i64 %v2)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
br i1 %obit, label %overflow, label %continue

overflow:
ret i1 false

continue:
ret i1 true
}

```

```

define zeroext i1 @usubo.br.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL: usubo.br.i32:

```

```

; RV64:    # %bb.0: # %entry
; RV64-NEXT:  subw a1, a0, a1
; RV64-NEXT:  bgeu a0, a1, .LBB58_2
; RV64-NEXT:  # %bb.1: # %overflow
; RV64-NEXT:  li a0, 0
; RV64-NEXT:  ret
; RV64-NEXT:  .LBB58_2: # %continue
; RV64-NEXT:  li a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.usub.with.overflow.i32(i32 %v1, i32 %v2)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
br i1 %obit, label %overflow, label %continue

overflow:
ret i1 false

continue:
ret i1 true
}

define zeroext i1 @usubo.br.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: usubo.br.i64:
;
RV64:    # %bb.0: # %entry
; RV64-NEXT:  sub a1, a0, a1
; RV64-NEXT:  bgeu a0, a1, .LBB59_2
; RV64-NEXT:  # %bb.1: # %overflow
; RV64-NEXT:  li a0, 0
; RV64-NEXT:  ret
; RV64-NEXT:  .LBB59_2: # %continue
; RV64-NEXT:  li a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.usub.with.overflow.i64(i64 %v1, i64 %v2)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
br i1 %obit, label %overflow, label %continue

overflow:
ret i1 false

continue:
ret i1 true
}

define zeroext i1 @smulo.br.i32(i32 signext %v1, i32 signext %v2) {

```

```

; RV64-LABEL: smulo.br.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  mul a0, a0, a1
; RV64-NEXT:  srai a1, a0, 32
; RV64-NEXT:  sraiw a0, a0, 31
; RV64-NEXT:  beq a1, a0, .LBB60_2
; RV64-NEXT:  # %bb.1: # %overflow
; RV64-NEXT:  li a0, 0
; RV64-NEXT:  ret
; RV64-NEXT: .LBB60_2: # %continue
; RV64-NEXT:  li a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i32, i1} @llvm.smul.with.overflow.i32(i32
%v1, i32 %v2)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
br i1 %obit, label %overflow, label %continue

overflow:
ret i1 false

continue:
ret i1 true
}

```

```

define zeroext i1 @smulo.br.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: smulo.br.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  mulh a2, a0, a1
; RV64-NEXT:  mul a0, a0, a1
; RV64-NEXT:  srai a0, a0, 63
; RV64-NEXT:  beq a2, a0, .LBB61_2
; RV64-NEXT:  # %bb.1: # %overflow
; RV64-NEXT:  li a0, 0
; RV64-NEXT:  ret
; RV64-NEXT: .LBB61_2: # %continue
; RV64-NEXT:  li a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.smul.with.overflow.i64(i64 %v1, i64 %v2)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
br i1 %obit, label %overflow, label %continue

overflow:
ret i1 false

```

```

continue:
    ret i1 true
}

define zeroext i1 @smulo2.br.i64(i64 %v1) {
; RV64-LABEL: smulo2.br.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  li a1, -13
; RV64-NEXT:
    mulh a2, a0, a1
; RV64-NEXT:  mul a0, a0, a1
; RV64-NEXT:  srai a0, a0, 63
; RV64-NEXT:  beq a2, a0, .LBB62_2
; RV64-NEXT:  # %bb.1: # %overflow
; RV64-NEXT:  li a0, 0
; RV64-NEXT:  ret
; RV64-NEXT: .LBB62_2: # %continue
; RV64-NEXT:  li a0, 1
; RV64-NEXT:  ret
entry:
    %t = call {i64, i1} @llvm.smul.with.overflow.i64(i64 %v1, i64 -13)
    %val = extractvalue {i64, i1} %t, 0
    %obit = extractvalue {i64, i1} %t, 1
    br i1 %obit, label %overflow, label %continue

overflow:
    ret i1 false

continue:
    ret i1 true
}

define zeroext i1 @umulo.br.i32(i32 signext %v1, i32 signext %v2) {
; RV64-LABEL: umulo.br.i32:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  slli a1, a1, 32
; RV64-NEXT:  slli a0, a0, 32
; RV64-NEXT:  mulhu a0, a0, a1
; RV64-NEXT:  srai a0, a0, 32
; RV64-NEXT:  beqz a0, .LBB63_2
; RV64-NEXT:  # %bb.1: # %overflow
; RV64-NEXT:  li a0, 0
; RV64-NEXT:  ret
; RV64-NEXT: .LBB63_2: # %continue
; RV64-NEXT:  li a0, 1
; RV64-NEXT:  ret
entry:
    %t = call

```

```

{i32, i1} @llvm.umul.with.overflow.i32(i32 %v1, i32 %v2)
%val = extractvalue {i32, i1} %t, 0
%obit = extractvalue {i32, i1} %t, 1
br i1 %obit, label %overflow, label %continue

overflow:
ret i1 false

continue:
ret i1 true
}

define zeroext i1 @umulo.br.i64(i64 %v1, i64 %v2) {
; RV64-LABEL: umulo.br.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  mulhu a0, a0, a1
; RV64-NEXT:  beqz a0, .LBB64_2
; RV64-NEXT:  # %bb.1: # %overflow
; RV64-NEXT:  li a0, 0
; RV64-NEXT:  ret
; RV64-NEXT: .LBB64_2: # %continue
; RV64-NEXT:  li a0, 1
; RV64-NEXT:  ret
entry:
%t = call {i64, i1} @llvm.umul.with.overflow.i64(i64 %v1, i64 %v2)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
br i1 %obit, label %overflow, label %continue

overflow:
ret i1 false

continue:
ret i1 true
}

define zeroext i1 @umulo2.br.i64(i64 %v1) {
; RV64-LABEL: umulo2.br.i64:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  add a1, a0, a0
; RV64-NEXT:  bgeu a1, a0,
.LBB65_2
; RV64-NEXT:  # %bb.1: # %overflow
; RV64-NEXT:  li a0, 0
; RV64-NEXT:  ret
; RV64-NEXT: .LBB65_2: # %continue
; RV64-NEXT:  li a0, 1
; RV64-NEXT:  ret

```



```

entry:
    %t = call {i64, i1} @llvm.umul.with.overflow.i64(i64 %v1, i64 2)
    %val = extractvalue {i64, i1} %t, 0
    %obit = extractvalue {i64, i1} %t, 1
    br i1 %obit, label %overflow, label %continue

overflow:
    ret i1 false

continue:
    ret i1 true
}

define zeroext i1 @uaddo.i64.constant(i64 %v1, ptr %res) {
; RV64-LABEL: uaddo.i64.constant:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  addi a2, a0, 2
; RV64-NEXT:  sltu a0, a2, a0
; RV64-NEXT:  sd a2, 0(a1)
; RV64-NEXT:  ret
entry:
    %t = call {i64, i1} @llvm.uadd.with.overflow.i64(i64 %v1, i64 2)
    %val = extractvalue {i64, i1} %t, 0
    %obit = extractvalue {i64, i1} %t, 1
    store i64 %val, ptr %res
    ret i1 %obit
}

define zeroext i1 @uaddo.i64.constant_2048(i64 %v1, ptr %res) {
; RV64-LABEL: uaddo.i64.constant_2048:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:
    addi a2, a0, 2047
; RV64-NEXT:  addi a2, a2, 1
; RV64-NEXT:  sltu a0, a2, a0
; RV64-NEXT:  sd a2, 0(a1)
; RV64-NEXT:  ret
entry:
    %t = call {i64, i1} @llvm.uadd.with.overflow.i64(i64 %v1, i64 2048)
    %val = extractvalue {i64, i1} %t, 0
    %obit = extractvalue {i64, i1} %t, 1
    store i64 %val, ptr %res
    ret i1 %obit
}

define zeroext i1 @uaddo.i64.constant_2049(i64 %v1, ptr %res) {
; RV64-LABEL: uaddo.i64.constant_2049:
; RV64:      # %bb.0: # %entry

```

```

; RV64-NEXT:  addi a2, a0, 2047
; RV64-NEXT:  addi a2, a2, 2
; RV64-NEXT:  sltu a0, a2, a0
; RV64-NEXT:  sd a2, 0(a1)
; RV64-NEXT:  ret

entry:
%t = call {i64, i1} @llvm.uadd.with.overflow.i64(i64 %v1, i64 2049)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
store i64 %val, ptr %res
ret i1 %obit
}

```

```

define i64 @uaddo.i64.constant_setcc_on_overflow_flag(ptr %p) {
; RV64-LABEL: uaddo.i64.constant_setcc_on_overflow_flag:
; RV64:      # %bb.0: # %entry
; RV64-NEXT:  ld a1, 0(a0)
;
; RV64-NEXT:  addi a0, a1, 2
; RV64-NEXT:  bltu a0, a1, .LBB69_2
; RV64-NEXT:  # %bb.1: # %IfOverflow
; RV64-NEXT:  li a0, 0
; RV64-NEXT:  .LBB69_2: # %IfNoOverflow
; RV64-NEXT:  ret

entry:
%v1 = load i64, ptr %p
%t = call {i64, i1} @llvm.uadd.with.overflow.i64(i64 %v1, i64 2)
%val = extractvalue {i64, i1} %t, 0
%obit = extractvalue {i64, i1} %t, 1
br i1 %obit, label %IfNoOverflow, label %IfOverflow
IfOverflow:
ret i64 0
IfNoOverflow:
ret i64 %val
}

```

```

declare {i32, i1} @llvm.sadd.with.overflow.i32(i32, i32) nounwind readnone
declare {i64, i1} @llvm.sadd.with.overflow.i64(i64, i64) nounwind readnone
declare {i32, i1} @llvm.uadd.with.overflow.i32(i32, i32) nounwind readnone
declare {i64, i1} @llvm.uadd.with.overflow.i64(i64, i64) nounwind readnone
declare {i32, i1} @llvm.ssub.with.overflow.i32(i32, i32) nounwind readnone
declare {i64, i1} @llvm.ssub.with.overflow.i64(i64, i64) nounwind readnone
declare {i32, i1} @llvm.usub.with.overflow.i32(i32, i32) nounwind readnone
declare
{i64, i1} @llvm.usub.with.overflow.i64(i64, i64) nounwind readnone
declare {i32, i1} @llvm.smul.with.overflow.i32(i32, i32) nounwind readnone
declare {i64, i1} @llvm.smul.with.overflow.i64(i64, i64) nounwind readnone
declare {i32, i1} @llvm.umul.with.overflow.i32(i32, i32) nounwind readnone

```

```
declare {i64, i1} @llvm.umul.with.overflow.i64(i64, i64) nounwind readnone
add_llvm_library(lldbPluginObjectContainerBSDArchive PLUGIN
ObjectContainerBSDArchive.cpp
```

LINK\_LIBS

lldbCore

lldbHost

lldbSymbol

LINK\_COMPONENTS

Support

)

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the

power, direct or indirect, to cause the

direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such

obligations, You may act only

on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

### ---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and

prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

=====

Software from third parties included in the LLVM Project:

=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or `LICENSE` file at the top containing the specific license and restrictions which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every file.

=====

Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):

=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2007-2019 University of Illinois at Urbana-Champaign.  
All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

<http://llvm.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do

so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.



\* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

```
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck -check-prefix=RV64I %s
; RUN: llc -mtriple=riscv64 -mattr=+m -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck -check-prefix=RV64IM %s
```

```
define i32 @udiv(i32 %a, i32 %b) nounwind {
; RV64I-LABEL: udiv:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  slli a1, a1, 32
; RV64I-NEXT:  srli a1, a1, 32
; RV64I-NEXT:  call __udivdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: udiv:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  divuw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = udiv i32 %a, %b
ret i32 %1
}
```

```
define i32 @udiv_constant(i32 %a) nounwind {
; RV64I-LABEL:
; udiv_constant:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
```

```

; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a0, a0, 32
; RV64I-NEXT:  li a1, 5
; RV64I-NEXT:  call __udivdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: udiv_constant:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  slli a0, a0, 32
; RV64IM-NEXT:  lui a1, 838861
; RV64IM-NEXT:  addi a1, a1, -819
; RV64IM-NEXT:  slli a1, a1, 32
; RV64IM-NEXT:  mulhu a0, a0, a1
; RV64IM-NEXT:  srli a0, a0, 34
; RV64IM-NEXT:  ret
%1 = udiv i32 %a, 5
ret i32 %1
}

```

```

define i32 @udiv_pow2(i32 %a) nounwind {
; RV64I-LABEL: udiv_pow2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  srliw a0, a0, 3
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: udiv_pow2:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  srliw a0, a0, 3
; RV64IM-NEXT:  ret
%1 = udiv i32 %a, 8
ret i32 %1
}

```

```

define i32 @udiv_constant_lhs(i32
%a) nounwind {
; RV64I-LABEL: udiv_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a0, a0, 32
; RV64I-NEXT:  srli a1, a0, 32
; RV64I-NEXT:  li a0, 10
; RV64I-NEXT:  call __udivdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret

```

```

;
; RV64IM-LABEL: udiv_constant_lhs:
; RV64IM:      # %bb.0:
; RV64IM-NEXT: li a1, 10
; RV64IM-NEXT: divuw a0, a1, a0
; RV64IM-NEXT: ret
%1 = udiv i32 10, %a
ret i32 %1
}

```

```

define i64 @udiv64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: udiv64:
; RV64I:      # %bb.0:
; RV64I-NEXT: tail __udivdi3
;
; RV64IM-LABEL: udiv64:
; RV64IM:      # %bb.0:
; RV64IM-NEXT: divu a0, a0, a1
; RV64IM-NEXT: ret
%1 = udiv i64 %a, %b
ret i64 %1
}

```

```

define i64 @udiv64_constant(i64 %a) nounwind {
; RV64I-LABEL: udiv64_constant:
; RV64I:      # %bb.0:
; RV64I-NEXT: li a1, 5
; RV64I-NEXT:
; tail __udivdi3
;
; RV64IM-LABEL: udiv64_constant:
; RV64IM:      # %bb.0:
; RV64IM-NEXT: lui a1, 838861
; RV64IM-NEXT: addiw a1, a1, -819
; RV64IM-NEXT: slli a2, a1, 32
; RV64IM-NEXT: add a1, a1, a2
; RV64IM-NEXT: mulhu a0, a0, a1
; RV64IM-NEXT: srli a0, a0, 2
; RV64IM-NEXT: ret
%1 = udiv i64 %a, 5
ret i64 %1
}

```

```

define i64 @udiv64_constant_lhs(i64 %a) nounwind {
; RV64I-LABEL: udiv64_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT: mv a1, a0
; RV64I-NEXT: li a0, 10

```

```

; RV64I-NEXT:  tail __udivdi3
;
; RV64IM-LABEL: udiv64_constant_lhs:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  li a1, 10
; RV64IM-NEXT:  divu a0, a1, a0
; RV64IM-NEXT:  ret
%1 = udiv i64 10, %a
ret i64 %1
}

define i8 @udiv8(i8 %a, i8 %b) nounwind {
; RV64I-LABEL: udiv8:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  andi a0, a0, 255
; RV64I-NEXT:  andi a1, a1, 255
; RV64I-NEXT:  call
__udivdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: udiv8:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  andi a1, a1, 255
; RV64IM-NEXT:  andi a0, a0, 255
; RV64IM-NEXT:  divuw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = udiv i8 %a, %b
ret i8 %1
}

define i8 @udiv8_constant(i8 %a) nounwind {
; RV64I-LABEL: udiv8_constant:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  andi a0, a0, 255
; RV64I-NEXT:  li a1, 5
; RV64I-NEXT:  call __udivdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: udiv8_constant:
; RV64IM:      # %bb.0:

```

```

; RV64IM-NEXT:  andi a0, a0, 255
; RV64IM-NEXT:  li a1, 205
; RV64IM-NEXT:  mul a0, a0, a1
; RV64IM-NEXT:  srliw a0, a0, 10
; RV64IM-NEXT:  ret
%1 = udiv i8 %a, 5
ret i8 %1
}

define i8 @udiv8_pow2(i8 %a) nounwind
{
; RV64I-LABEL: udiv8_pow2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 56
; RV64I-NEXT:  srli a0, a0, 59
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: udiv8_pow2:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  slli a0, a0, 56
; RV64IM-NEXT:  srli a0, a0, 59
; RV64IM-NEXT:  ret
%1 = udiv i8 %a, 8
ret i8 %1
}

define i8 @udiv8_constant_lhs(i8 %a) nounwind {
; RV64I-LABEL: udiv8_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  andi a1, a0, 255
; RV64I-NEXT:  li a0, 10
; RV64I-NEXT:  call __udivdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: udiv8_constant_lhs:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  andi a0, a0, 255
; RV64IM-NEXT:  li a1, 10
; RV64IM-NEXT:  divuw a0, a1, a0
; RV64IM-NEXT:  ret
%1 = udiv i8 10, %a
ret i8 %1
}

```

```

define i16 @udiv16(i16 %a, i16 %b) nounwind {
; RV64I-LABEL: udiv16:
;
; RV64I:    # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  lui a2, 16
; RV64I-NEXT:  addiw a2, a2, -1
; RV64I-NEXT:  and a0, a0, a2
; RV64I-NEXT:  and a1, a1, a2
; RV64I-NEXT:  call __udivdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: udiv16:
; RV64IM:    # %bb.0:
; RV64IM-NEXT:  lui a2, 16
; RV64IM-NEXT:  addi a2, a2, -1
; RV64IM-NEXT:  and a1, a1, a2
; RV64IM-NEXT:  and a0, a0, a2
; RV64IM-NEXT:  divuw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = udiv i16 %a, %b
ret i16 %1
}

```

```

define i16 @udiv16_constant(i16 %a) nounwind {
; RV64I-LABEL: udiv16_constant:
; RV64I:    # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srli a0, a0, 48
; RV64I-NEXT:  li a1, 5
; RV64I-NEXT:  call __udivdi3
; RV64I-NEXT:  ld ra, 8(sp)
; RV64I-NEXT:  # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: udiv16_constant:
; RV64IM:    # %bb.0:
; RV64IM-NEXT:  lui a1, 52429
; RV64IM-NEXT:  slli a1, a1, 4
; RV64IM-NEXT:  slli a0, a0, 48
; RV64IM-NEXT:  mulhu a0, a0, a1
; RV64IM-NEXT:  srliw a0, a0, 18

```

```

; RV64IM-NEXT:  ret
%1 = udiv i16 %a, 5
ret i16 %1
}

define i16 @udiv16_pow2(i16 %a) nounwind {
; RV64I-LABEL: udiv16_pow2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srli a0, a0, 51
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: udiv16_pow2:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  slli a0, a0, 48
; RV64IM-NEXT:  srli a0, a0, 51
; RV64IM-NEXT:  ret
%1 = udiv i16 %a, 8
ret i16 %1
}

define i16 @udiv16_constant_lhs(i16 %a) nounwind {
; RV64I-LABEL: udiv16_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a0, a0, 48
; RV64I-NEXT:  srli a1, a0, 48
; RV64I-NEXT:
    li a0, 10
; RV64I-NEXT:  call __udivdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: udiv16_constant_lhs:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  slli a0, a0, 48
; RV64IM-NEXT:  srli a0, a0, 48
; RV64IM-NEXT:  li a1, 10
; RV64IM-NEXT:  divuw a0, a1, a0
; RV64IM-NEXT:  ret
%1 = udiv i16 10, %a
ret i16 %1
}

define i32 @sdiv(i32 %a, i32 %b) nounwind {
; RV64I-LABEL: sdiv:

```

```

; RV64I:    # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  sext.w a0, a0
; RV64I-NEXT:  sext.w a1, a1
; RV64I-NEXT:  call __divdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: sdiv:
; RV64IM:    # %bb.0:
; RV64IM-NEXT:  divw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = sdiv i32 %a, %b
ret i32 %1
}

```

```

define i32 @sdiv_constant(i32 %a) nounwind {
; RV64I-LABEL: sdiv_constant:
;
; RV64I:    # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  sext.w a0, a0
; RV64I-NEXT:  li a1, 5
; RV64I-NEXT:  call __divdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: sdiv_constant:
; RV64IM:    # %bb.0:
; RV64IM-NEXT:  sext.w a0, a0
; RV64IM-NEXT:  lui a1, 419430
; RV64IM-NEXT:  addiw a1, a1, 1639
; RV64IM-NEXT:  mul a0, a0, a1
; RV64IM-NEXT:  srli a1, a0, 63
; RV64IM-NEXT:  srai a0, a0, 33
; RV64IM-NEXT:  addw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = sdiv i32 %a, 5
ret i32 %1
}

```

```

define i32 @sdiv_pow2(i32 %a) nounwind {
; RV64I-LABEL: sdiv_pow2:
; RV64I:    # %bb.0:

```



```

; RV64I-NEXT:  sraiw a1, a0, 31
; RV64I-NEXT:  srlw a1, a1, 29
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  sraiw a0, a0, 3
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: sdiv_pow2:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  sraiw a1, a0,
31
; RV64IM-NEXT:  srlw a1, a1, 29
; RV64IM-NEXT:  add a0, a0, a1
; RV64IM-NEXT:  sraiw a0, a0, 3
; RV64IM-NEXT:  ret
%1 = sdiv i32 %a, 8
ret i32 %1
}

```

```

define i32 @sdiv_pow2_2(i32 %a) nounwind {
; RV64I-LABEL: sdiv_pow2_2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  sraiw a1, a0, 31
; RV64I-NEXT:  srlw a1, a1, 16
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  sraiw a0, a0, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: sdiv_pow2_2:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  sraiw a1, a0, 31
; RV64IM-NEXT:  srlw a1, a1, 16
; RV64IM-NEXT:  add a0, a0, a1
; RV64IM-NEXT:  sraiw a0, a0, 16
; RV64IM-NEXT:  ret
%1 = sdiv i32 %a, 65536
ret i32 %1
}

```

```

define i32 @sdiv_constant_lhs(i32 %a) nounwind {
; RV64I-LABEL: sdiv_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  sext.w a1, a0
; RV64I-NEXT:  li a0, -10
; RV64I-NEXT:  call __divdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded

```

```

Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: sdiv_constant_lhs:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  li a1, -10
; RV64IM-NEXT:  divw a0, a1, a0
; RV64IM-NEXT:  ret
%1 = sdiv i32 -10, %a
ret i32 %1
}

```

```

define i64 @sdiv64(i64 %a, i64 %b) nounwind {
; RV64I-LABEL: sdiv64:
; RV64I:      # %bb.0:
; RV64I-NEXT:  tail __divdi3
;
; RV64IM-LABEL: sdiv64:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  div a0, a0, a1
; RV64IM-NEXT:  ret
%1 = sdiv i64 %a, %b
ret i64 %1
}

```

```

define i64 @sdiv64_constant(i64 %a) nounwind {
; RV64I-LABEL: sdiv64_constant:
; RV64I:      # %bb.0:
; RV64I-NEXT:  li a1, 5
; RV64I-NEXT:  tail __divdi3
;
; RV64IM-LABEL: sdiv64_constant:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  lui a1, %hi(.LCPI21_0)
; RV64IM-NEXT:  ld a1, %lo(.LCPI21_0)(a1)
; RV64IM-NEXT:  mulh a0, a0, a1
; RV64IM-NEXT:  srli a1, a0, 63
; RV64IM-NEXT:  srai a0, a0, 1
; RV64IM-NEXT:  add a0, a0, a1
; RV64IM-NEXT:  ret
%1 = sdiv i64 %a,
5
ret i64 %1
}

```

```

define i64 @sdiv64_constant_lhs(i64 %a) nounwind {
; RV64I-LABEL: sdiv64_constant_lhs:

```

```

; RV64I:    # %bb.0:
; RV64I-NEXT:  mv a1, a0
; RV64I-NEXT:  li a0, 10
; RV64I-NEXT:  tail __divdi3
;
; RV64IM-LABEL: sdiv64_constant_lhs:
; RV64IM:    # %bb.0:
; RV64IM-NEXT:  li a1, 10
; RV64IM-NEXT:  div a0, a1, a0
; RV64IM-NEXT:  ret
%1 = sdiv i64 10, %a
ret i64 %1
}

```

; Although this sdiv has two sexti32 operands, it shouldn't compile to divw on  
; RV64M as that wouldn't produce the correct result for e.g. INT\_MIN/-1.

```

define i64 @sdiv64_sext_operands(i32 %a, i32 %b) nounwind {
; RV64I-LABEL: sdiv64_sext_operands:
; RV64I:    # %bb.0:
; RV64I-NEXT:  sext.w a0, a0
; RV64I-NEXT:  sext.w a1, a1
; RV64I-NEXT:  tail __divdi3
;
; RV64IM-LABEL: sdiv64_sext_operands:
; RV64IM:    # %bb.0:
; RV64IM-NEXT:  sext.w a0, a0
; RV64IM-NEXT:  sext.w a1, a1
; RV64IM-NEXT:  div a0, a0, a1
; RV64IM-NEXT:  ret
%1 = sext i32 %a to i64
%2 = sext i32 %b to
i64
%3 = sdiv i64 %1, %2
ret i64 %3
}

```

```

define i8 @sdiv8(i8 %a, i8 %b) nounwind {
; RV64I-LABEL: sdiv8:
; RV64I:    # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a1, a1, 24
; RV64I-NEXT:  sraiw a1, a1, 24
; RV64I-NEXT:  slli a0, a0, 24
; RV64I-NEXT:  sraiw a0, a0, 24
; RV64I-NEXT:  call __divdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload

```

```

; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: sdiv8:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  slli a1, a1, 24
; RV64IM-NEXT:  sraiw a1, a1, 24
; RV64IM-NEXT:  slli a0, a0, 24
; RV64IM-NEXT:  sraiw a0, a0, 24
; RV64IM-NEXT:  divw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = sdiv i8 %a, %b
ret i8 %1
}

define i8 @sdiv8_constant(i8 %a) nounwind {
; RV64I-LABEL: sdiv8_constant:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a0, a0, 24
; RV64I-NEXT:
    sraiw a0, a0, 24
; RV64I-NEXT:  li a1, 5
; RV64I-NEXT:  call __divdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: sdiv8_constant:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  slli a0, a0, 24
; RV64IM-NEXT:  sraiw a0, a0, 24
; RV64IM-NEXT:  li a1, 103
; RV64IM-NEXT:  mul a0, a0, a1
; RV64IM-NEXT:  sraiw a1, a0, 9
; RV64IM-NEXT:  slli a0, a0, 48
; RV64IM-NEXT:  srli a0, a0, 63
; RV64IM-NEXT:  addw a0, a1, a0
; RV64IM-NEXT:  ret
%1 = sdiv i8 %a, 5
ret i8 %1
}

define i8 @sdiv8_pow2(i8 %a) nounwind {
; RV64I-LABEL: sdiv8_pow2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a1, a0, 24

```

```

; RV64I-NEXT:  sraiw a1, a1, 24
; RV64I-NEXT:  slli a1, a1, 49
; RV64I-NEXT:  srli a1, a1, 61
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  slli a0, a0, 24
; RV64I-NEXT:  sraiw a0, a0, 27
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: sdiv8_pow2:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  slli a1, a0,
24
; RV64IM-NEXT:  sraiw a1, a1, 24
; RV64IM-NEXT:  slli a1, a1, 49
; RV64IM-NEXT:  srli a1, a1, 61
; RV64IM-NEXT:  add a0, a0, a1
; RV64IM-NEXT:  slli a0, a0, 24
; RV64IM-NEXT:  sraiw a0, a0, 27
; RV64IM-NEXT:  ret
%1 = sdiv i8 %a, 8
ret i8 %1
}

define i8 @sdiv8_constant_lhs(i8 %a) nounwind {
; RV64I-LABEL: sdiv8_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a0, a0, 24
; RV64I-NEXT:  sraiw a1, a0, 24
; RV64I-NEXT:  li a0, -10
; RV64I-NEXT:  call __divdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: sdiv8_constant_lhs:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  slli a0, a0, 24
; RV64IM-NEXT:  sraiw a0, a0, 24
; RV64IM-NEXT:  li a1, -10
; RV64IM-NEXT:  divw a0, a1, a0
; RV64IM-NEXT:  ret
%1 = sdiv i8 -10, %a
ret i8 %1
}

define i16 @sdiv16(i16 %a, i16 %b) nounwind {

```

```

; RV64I-LABEL:
sdiv16:
; RV64I:    # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a1, a1, 16
; RV64I-NEXT:  sraiw a1, a1, 16
; RV64I-NEXT:  slli a0, a0, 16
; RV64I-NEXT:  sraiw a0, a0, 16
; RV64I-NEXT:  call __divdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: sdiv16:
; RV64IM:    # %bb.0:
; RV64IM-NEXT:  slli a1, a1, 16
; RV64IM-NEXT:  sraiw a1, a1, 16
; RV64IM-NEXT:  slli a0, a0, 16
; RV64IM-NEXT:  sraiw a0, a0, 16
; RV64IM-NEXT:  divw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = sdiv i16 %a, %b
ret i16 %1
}

```

```

define i16 @sdiv16_constant(i16 %a) nounwind {
; RV64I-LABEL: sdiv16_constant:
; RV64I:    # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a0, a0, 16
; RV64I-NEXT:  sraiw a0, a0, 16
; RV64I-NEXT:  li a1, 5
; RV64I-NEXT:  call __divdi3
; RV64I-NEXT:
    ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: sdiv16_constant:
; RV64IM:    # %bb.0:
; RV64IM-NEXT:  slli a0, a0, 16
; RV64IM-NEXT:  sraiw a0, a0, 16
; RV64IM-NEXT:  lui a1, 6
; RV64IM-NEXT:  addi a1, a1, 1639
; RV64IM-NEXT:  mul a0, a0, a1
; RV64IM-NEXT:  srliw a1, a0, 31

```

```

; RV64IM-NEXT:  sraiw a0, a0, 17
; RV64IM-NEXT:  addw a0, a0, a1
; RV64IM-NEXT:  ret
%1 = sdiv i16 %a, 5
ret i16 %1
}

```

```

define i16 @sdiv16_pow2(i16 %a) nounwind {
; RV64I-LABEL: sdiv16_pow2:
; RV64I:      # %bb.0:
; RV64I-NEXT:  slli a1, a0, 16
; RV64I-NEXT:  sraiw a1, a1, 16
; RV64I-NEXT:  slli a1, a1, 33
; RV64I-NEXT:  srli a1, a1, 61
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  slli a0, a0, 16
; RV64I-NEXT:  sraiw a0, a0, 19
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: sdiv16_pow2:
; RV64IM:      # %bb.0:
; RV64IM-NEXT:  slli a1, a0, 16
; RV64IM-NEXT:  sraiw a1, a1, 16
; RV64IM-NEXT:  slli a1, a1, 33
;
; RV64IM-NEXT:  srli a1, a1, 61
; RV64IM-NEXT:  add a0, a0, a1
; RV64IM-NEXT:  slli a0, a0, 16
; RV64IM-NEXT:  sraiw a0, a0, 19
; RV64IM-NEXT:  ret
%1 = sdiv i16 %a, 8
ret i16 %1
}

```

```

define i16 @sdiv16_constant_lhs(i16 %a) nounwind {
; RV64I-LABEL: sdiv16_constant_lhs:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  slli a0, a0, 16
; RV64I-NEXT:  sraiw a1, a0, 16
; RV64I-NEXT:  li a0, -10
; RV64I-NEXT:  call __divdi3
; RV64I-NEXT:  ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
;
; RV64IM-LABEL: sdiv16_constant_lhs:

```

```
; RV64IM:    # %bb.0:
; RV64IM-NEXT:  slli a0, a0, 16
; RV64IM-NEXT:  sraiw a0, a0, 16
; RV64IM-NEXT:  li a1, -10
; RV64IM-NEXT:  divw a0, a1, a0
; RV64IM-NEXT:  ret
%1 = sdiv i16 -10, %a
ret i16 %1
}
```

LLVM System Interface Library

-----

Part of the LLVM Project, under the Apache License v2.0 with LLVM Exceptions.

See <https://llvm.org/LICENSE.txt> for license information.

SPDX-License-Identifier: Apache-2.0 WITH LLVM-exception

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.



"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional

attribution notices cannot be construed  
as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing

the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

### ---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled

forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

=====  
Software from third parties included in the LLVM Project:  
=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or `LICENSE` file at the top containing the specific license and restrictions which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every file.

=====  
Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):  
=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2011-2019 by the contributors listed in CREDITS.TXT  
All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

<http://llvm.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

\* Redistributions of source code must retain the above copyright notice,

this list of conditions and the following disclaimers.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.

\* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the

power, direct or indirect, to cause the

direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative

Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents



of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

## APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## ---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you

may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

=====  
Software from third parties included in the LLVM Project:  
=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or `LICENSE` file at the top containing the specific license and restrictions which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every file.

=====  
Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):  
=====

The software contained in this directory tree is dual licensed under both the University of Illinois "BSD-Like" license and the MIT license. As a user of this code you may choose to use it under either license. As a contributor, you agree to allow your code to be used under both. The full text of the relevant licenses is included below.

In addition, a license agreement from the copyright/patent holders of the software contained in this directory tree is included below.

=====  
University of Illinois/NCSA  
Open Source License

Copyright  
(c) 1997-2019 Intel Corporation

All rights reserved.

Developed by:

OpenMP Runtime Team

Intel Corporation

<http://www.openmp.rtl.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- \* Neither the names of Intel Corporation OpenMP Runtime Team nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====

Copyright (c) 1997-2019 Intel Corporation

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

Intel Corporation

Software Grant License Agreement ("Agreement")

Except for the license granted herein to you, Intel Corporation ("Intel") reserves all right, title, and interest in and to the Software (defined below).

#### Definition

"Software" means the code and documentation as well as any original work of authorship, including any modifications or additions to an existing work, that is intentionally submitted by Intel to [llvm.org](http://llvm.org) (<http://llvm.org>) ("LLVM") for inclusion in, or documentation of, any of the products owned or managed by LLVM (the "Work"). For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to LLVM or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, LLVM for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked otherwise.

1. Grant of Copyright License. Subject to the terms and conditions of this Agreement, Intel hereby grants to you and to recipients of the Software distributed by LLVM a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, sublicense, and distribute the Software and such derivative works.
2. Grant of Patent License. Subject to the terms and conditions of this Agreement, Intel hereby grants you and to recipients of the Software distributed by LLVM a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by Intel that are necessarily infringed by Intel's Software alone or by combination of the Software with the Work to which such Software was submitted. If any entity institutes patent litigation against Intel or any

other entity (including  
a cross-claim or counterclaim in a lawsuit) alleging  
that Intel's Software, or the Work to which Intel has contributed constitutes  
direct or contributory patent infringement, then any patent licenses granted  
to that entity under this Agreement for the Software or Work shall terminate  
as of the date such litigation is filed.

Unless required by applicable law or agreed to in writing, the software is  
provided on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,  
either express or implied, including, without limitation, any warranties or  
conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A  
PARTICULAR PURPOSE.

=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,  
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by  
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all  
other entities that control, are controlled by, or are under common  
control with that entity. For the purposes of this definition,  
"control" means (i) the power, direct or indirect, to cause the  
direction or management of such entity, whether by contract or  
otherwise, or (ii) ownership of fifty percent (50%) or more of the  
outstanding shares, or (iii) beneficial ownership  
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity  
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,  
including but not limited to software source code, documentation  
source, and configuration files.

"Object" form shall mean any form resulting from mechanical  
transformation or translation of a Source form, including but  
not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions



for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only

on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the

Licensors for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

## APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

=====  
Software from third parties included in the LLVM Project:  
=====

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or `LICENSE` file at the top containing the specific license and restrictions which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every file.

=====  
Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):  
=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2009-2019 Polly Team  
All rights reserved.

Developed by:

Polly Team

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- \* Neither the names of the Polly Team, copyright holders, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

=====

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all



other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative

Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work

includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional

attribution notices cannot be construed  
as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions  
of this Software are embedded into an Object form of such source code, you  
may redistribute such embedded portions in such Object form without complying  
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled  
forms of this Software with  
software that is licensed under the GPLv2 ("Combined Software") and if a  
court of competent jurisdiction determines that the patent provision (Section  
3), the indemnity provision (Section 9) or other Section of the License  
conflicts with the conditions of the GPLv2, you may retroactively and  
prospectively choose to deem waived or otherwise exclude such Section(s) of  
the License, but only in their entirety and only with respect to the Combined  
Software.

=====  
Software from third parties included in the LLVM Project:  
=====

The LLVM Project contains third party software which is under different license  
terms. All such code will be identified clearly using at least one of two  
mechanisms:

- 1) It will be in a separate directory tree with its own `LICENSE.txt` or  
`LICENSE` file at the top containing the specific  
license and restrictions  
which apply to that software, or
- 2) It will contain specific license and restriction terms at the top of every  
file.

=====  
Legacy LLVM License (<https://llvm.org/docs/DeveloperPolicy.html#legacy>):  
=====

The compiler\_rt library is dual licensed under both the University of Illinois  
"BSD-Like" license and the MIT license. As a user of this code you may choose  
to use it under either license. As a contributor, you agree to allow your code  
to be used under both.

Full text of the relevant licenses is included below.

=====

University of Illinois/NCSA  
Open Source License

Copyright (c) 2009-2019 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

<http://llvm.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- \* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

=====

Copyright (c) 2009-2015 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck -check-prefix=RV64I %s
```

```
; Check indexed and unindexed, sext, zext and anyext loads
```

```
define dso_local i64 @lb(ptr %a) nounwind {
; RV64I-LABEL: lb:
; RV64I:      # %bb.0:
; RV64I-NEXT: lb a1, 1(a0)
; RV64I-NEXT: lbu zero, 0(a0)
; RV64I-NEXT: mv a0, a1
; RV64I-NEXT: ret
%1 = getelementptr i8, ptr %a, i32 1
%2 = load i8, ptr %1
%3 = sext i8 %2 to i64
; the unused load will produce an anyext for selection
%4 = load volatile i8, ptr %a
ret i64 %3
}
```

```
define dso_local i64 @lh(ptr %a) nounwind {
; RV64I-LABEL: lh:
; RV64I:      # %bb.0:
; RV64I-NEXT: lh a1, 4(a0)
; RV64I-NEXT: lh zero, 0(a0)
; RV64I-NEXT: mv a0, a1
; RV64I-NEXT: ret
%1 = getelementptr i16, ptr %a, i32 2
```

```

%2 = load i16, ptr %1
%3 = sext i16 %2 to i64
; the unused load will produce an anyext for selection
%4 = load volatile i16, ptr %a
ret i64 %3
}

```

```

define dso_local i64 @lw(ptr %a) nounwind {
; RV64I-LABEL: lw:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lw a1, 12(a0)
; RV64I-NEXT:  lw zero, 0(a0)
; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT:  ret
%1 = getelementptr i32, ptr %a, i32 3
%2 = load i32, ptr %1
%3 = sext i32 %2 to i64
; the unused load will produce an anyext for selection
%4 = load volatile i32, ptr %a
ret i64 %3
}

```

```

define dso_local i64 @lbu(ptr %a) nounwind {
; RV64I-LABEL: lbu:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lbu a1, 4(a0)
; RV64I-NEXT:  lbu a0, 0(a0)
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  ret
%1 = getelementptr i8, ptr %a, i32 4
%2 = load i8, ptr %1
%3 = zext i8 %2 to i64
%4 = load volatile i8, ptr %a
%5 = zext i8 %4 to i64
%6 = add i64 %3, %5
ret i64 %6
}

```

```

define dso_local i64 @lhu(ptr %a) nounwind {
; RV64I-LABEL: lhu:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lhu a1, 10(a0)
; RV64I-NEXT:  lhu a0, 0(a0)
; RV64I-NEXT:
    add a0, a1, a0
; RV64I-NEXT:  ret
%1 = getelementptr i16, ptr %a, i32 5
%2 = load i16, ptr %1

```



```

%3 = zext i16 %2 to i64
%4 = load volatile i16, ptr %a
%5 = zext i16 %4 to i64
%6 = add i64 %3, %5
ret i64 %6
}

define dso_local i64 @lwu(ptr %a) nounwind {
; RV64I-LABEL: lwu:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lwu a1, 24(a0)
; RV64I-NEXT:  lwu a0, 0(a0)
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  ret
%1 = getelementptr i32, ptr %a, i32 6
%2 = load i32, ptr %1
%3 = zext i32 %2 to i64
%4 = load volatile i32, ptr %a
%5 = zext i32 %4 to i64
%6 = add i64 %3, %5
ret i64 %6
}

```

; 64-bit loads and stores

```

define dso_local i64 @ld(ptr %a) nounwind {
; RV64I-LABEL: ld:
; RV64I:      # %bb.0:
; RV64I-NEXT:  ld a1, 80(a0)
; RV64I-NEXT:  ld zero, 0(a0)
; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT:  ret
%1 = getelementptr i64, ptr %a, i32 10
%2 = load i64, ptr %1
%3 = load volatile i64, ptr %a
ret i64 %2
}

```

```

define dso_local void @sd(ptr %a, i64 %b)
nounwind {
; RV64I-LABEL: sd:
; RV64I:      # %bb.0:
; RV64I-NEXT:  sd a1, 0(a0)
; RV64I-NEXT:  sd a1, 88(a0)
; RV64I-NEXT:  ret
store i64 %b, ptr %a
%1 = getelementptr i64, ptr %a, i32 11
store i64 %b, ptr %1
}

```

```

ret void
}

; Check load and store to an i1 location
define dso_local i64 @load_sext_zext_anyext_i1(ptr %a) nounwind {
; RV64I-LABEL: load_sext_zext_anyext_i1:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lbu a1, 1(a0)
; RV64I-NEXT:  lbu a2, 2(a0)
; RV64I-NEXT:  lbu zero, 0(a0)
; RV64I-NEXT:  sub a0, a2, a1
; RV64I-NEXT:  ret
; sextload i1
%1 = getelementptr i1, ptr %a, i32 1
%2 = load i1, ptr %1
%3 = sext i1 %2 to i64
; zextload i1
%4 = getelementptr i1, ptr %a, i32 2
%5 = load i1, ptr %4
%6 = zext i1 %5 to i64
%7 = add i64 %3, %6
; extload i1 (anyext). Produced as the load is unused.
%8 = load volatile i1, ptr %a
ret i64 %7
}

define dso_local i16 @load_sext_zext_anyext_i1_i16(ptr %a) nounwind {
; RV64I-LABEL: load_sext_zext_anyext_i1_i16:
;
; RV64I:      # %bb.0:
; RV64I-NEXT:  lbu a1, 1(a0)
; RV64I-NEXT:  lbu a2, 2(a0)
; RV64I-NEXT:  lbu zero, 0(a0)
; RV64I-NEXT:  subw a0, a2, a1
; RV64I-NEXT:  ret
; sextload i1
%1 = getelementptr i1, ptr %a, i32 1
%2 = load i1, ptr %1
%3 = sext i1 %2 to i16
; zextload i1
%4 = getelementptr i1, ptr %a, i32 2
%5 = load i1, ptr %4
%6 = zext i1 %5 to i16
%7 = add i16 %3, %6
; extload i1 (anyext). Produced as the load is unused.
%8 = load volatile i1, ptr %a
ret i16 %7
}

```

```

; Check load and store to a global
@G = dso_local global i64 0

define dso_local i64 @ld_sd_global(i64 %a) nounwind {
; RV64I-LABEL: ld_sd_global:
; RV64I:      # %bb.0:
; RV64I-NEXT:  lui a2, %hi(G)
; RV64I-NEXT:  ld a1, %lo(G)(a2)
; RV64I-NEXT:  addi a3, a2, %lo(G)
; RV64I-NEXT:  sd a0, %lo(G)(a2)
; RV64I-NEXT:  ld zero, 72(a3)
; RV64I-NEXT:  sd a0, 72(a3)
; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT:  ret
%1 = load volatile i64, ptr @G
store i64 %a, ptr @G
%2 = getelementptr
i64, ptr @G, i64 9
%3 = load volatile i64, ptr %2
store i64 %a, ptr %2
ret i64 %1
}

define i64 @lw_near_local(ptr %a) {
; RV64I-LABEL: lw_near_local:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi a0, a0, 2047
; RV64I-NEXT:  ld a0, 9(a0)
; RV64I-NEXT:  ret
%1 = getelementptr inbounds i64, ptr %a, i64 257
%2 = load volatile i64, ptr %1
ret i64 %2
}

define void @st_near_local(ptr %a, i64 %b) {
; RV64I-LABEL: st_near_local:
; RV64I:      # %bb.0:
; RV64I-NEXT:  addi a0, a0, 2047
; RV64I-NEXT:  sd a1, 9(a0)
; RV64I-NEXT:  ret
%1 = getelementptr inbounds i64, ptr %a, i64 257
store i64 %b, ptr %1
ret void
}

define i64 @lw_sw_near_local(ptr %a, i64 %b) {
; RV64I-LABEL: lw_sw_near_local:

```

```

; RV64I:    # %bb.0:
; RV64I-NEXT:  addi a2, a0, 2047
; RV64I-NEXT:  ld a0, 9(a2)
; RV64I-NEXT:  sd a1, 9(a2)
; RV64I-NEXT:  ret
%1 = getelementptr inbounds i64, ptr %a, i64 257
%2 = load volatile i64, ptr %1
store i64 %b, ptr %1
ret i64 %2
}

```

```

define
i64 @lw_far_local(ptr %a) {
; RV64I-LABEL: lw_far_local:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a1, 8
; RV64I-NEXT:  add a0, a0, a1
; RV64I-NEXT:  ld a0, -8(a0)
; RV64I-NEXT:  ret
%1 = getelementptr inbounds i64, ptr %a, i64 4095
%2 = load volatile i64, ptr %1
ret i64 %2
}

```

```

define void @st_far_local(ptr %a, i64 %b) {
; RV64I-LABEL: st_far_local:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a2, 8
; RV64I-NEXT:  add a0, a0, a2
; RV64I-NEXT:  sd a1, -8(a0)
; RV64I-NEXT:  ret
%1 = getelementptr inbounds i64, ptr %a, i64 4095
store i64 %b, ptr %1
ret void
}

```

```

define i64 @lw_sw_far_local(ptr %a, i64 %b) {
; RV64I-LABEL: lw_sw_far_local:
; RV64I:    # %bb.0:
; RV64I-NEXT:  lui a2, 8
; RV64I-NEXT:  add a2, a0, a2
; RV64I-NEXT:  ld a0, -8(a2)
; RV64I-NEXT:  sd a1, -8(a2)
; RV64I-NEXT:  ret
%1 = getelementptr inbounds i64, ptr %a, i64 4095
%2 = load volatile i64, ptr %1
store i64 %b, ptr %1
ret i64 %2
}

```

```
}
```

```
; Make sure we don't fold the addiw  
into the load offset. The sign extend of the  
; addiw is required.
```

```
define i64 @lw_really_far_local(ptr %a) {  
; RV64I-LABEL: lw_really_far_local:  
; RV64I:    # %bb.0:  
; RV64I-NEXT:  lui a1, 524288  
; RV64I-NEXT:  addiw a1, a1, -2048  
; RV64I-NEXT:  add a0, a0, a1  
; RV64I-NEXT:  ld a0, 0(a0)  
; RV64I-NEXT:  ret  
%1 = getelementptr inbounds i64, ptr %a, i64 268435200  
%2 = load volatile i64, ptr %1  
ret i64 %2  
}
```

```
; Make sure we don't fold the addiw into the store offset. The sign extend of  
; the addiw is required.
```

```
define void @st_really_far_local(ptr %a, i64 %b) {  
; RV64I-LABEL: st_really_far_local:  
; RV64I:    # %bb.0:  
; RV64I-NEXT:  lui a2, 524288  
; RV64I-NEXT:  addiw a2, a2, -2048  
; RV64I-NEXT:  add a0, a0, a2  
; RV64I-NEXT:  sd a1, 0(a0)  
; RV64I-NEXT:  ret  
%1 = getelementptr inbounds i64, ptr %a, i64 268435200  
store i64 %b, ptr %1  
ret void  
}
```

```
; Make sure we don't fold the addiw into the load/store offset. The sign extend  
; of the addiw is required.
```

```
define  
i64 @lw_sw_really_far_local(ptr %a, i64 %b) {  
; RV64I-LABEL: lw_sw_really_far_local:  
; RV64I:    # %bb.0:  
; RV64I-NEXT:  lui a2, 524288  
; RV64I-NEXT:  addiw a2, a2, -2048  
; RV64I-NEXT:  add a2, a0, a2  
; RV64I-NEXT:  ld a0, 0(a2)  
; RV64I-NEXT:  sd a1, 0(a2)  
; RV64I-NEXT:  ret  
%1 = getelementptr inbounds i64, ptr %a, i64 268435200  
%2 = load volatile i64, ptr %1  
store i64 %b, ptr %1
```

```
ret i64 %2
}
```

```
%struct.quux = type { i32, [0 x i8] }
```

```
; Make sure we don't remove the addi and fold the C from
; (add (addi FrameIndex, C), X) into the store address.
; FrameIndex cannot be the operand of an ADD. We must keep the ADDI.
define void @addi_fold_crash(i64 %arg) nounwind {
; RV64I-LABEL: addi_fold_crash:
; RV64I:      # %bb.0: # %bb
; RV64I-NEXT:  addi sp, sp, -16
; RV64I-NEXT:  sd ra, 8(sp) # 8-byte Folded Spill
; RV64I-NEXT:  addi a1, sp, 4
; RV64I-NEXT:  add a0, a1, a0
; RV64I-NEXT:  sb zero, 0(a0)
; RV64I-NEXT:  mv a0, a1
; RV64I-NEXT:  call snork
; RV64I-NEXT:
    ld ra, 8(sp) # 8-byte Folded Reload
; RV64I-NEXT:  addi sp, sp, 16
; RV64I-NEXT:  ret
bb:
    %tmp = alloca %struct.quux, align 8
    %tmp1 = getelementptr inbounds %struct.quux, ptr %tmp, i64 0, i32 1
    %tmp2 = getelementptr inbounds %struct.quux, ptr %tmp, i64 0, i32 1, i64 %arg
    store i8 0, ptr %tmp2, align 1
    call void @snork(ptr %tmp1)
    ret void
}
```

```
declare void @snork(ptr)
; NOTE: Assertions have been autogenerated by utils/update_test_checks.py UTC_ARGS: --function-signature --
scrub-attributes
; RUN: opt -S -passes=argpromotion < %s | FileCheck %s
; Test that we only promote arguments when the caller/callee have compatible
; function attributes.
```

```
target triple = "x86_64-unknown-linux-gnu"
```

```
; This should promote
define internal fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512(ptr %arg, ptr
readonly %arg1) #0 {
; CHECK-LABEL: define {{{^@}+}}@callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512
; CHECK-SAME: (ptr [[ARG:%.*]], <8 x i64> [[ARG1_VAL:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT:  store <8 x i64> [[ARG1_VAL]], ptr [[ARG]]
; CHECK-NEXT:  ret void
```

```

;
bb:
    %tmp = load <8 x i64>, ptr %arg1
    store <8 x i64> %tmp, ptr %arg
    ret void
}

define void @avx512_legal512_prefer512_call_avx512_legal512_prefer512(ptr %arg) #0 {
; CHECK-LABEL: define {{{^@}+}}@avx512_legal512_prefer512_call_avx512_legal512_prefer512
; CHECK-SAME:
    (ptr [[ARG:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: [[TMP2:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: call void @llvm.memset.p0.i64(ptr align 32 [[TMP]], i8 0, i64 32, i1 false)
; CHECK-NEXT: [[TMP_VAL:%.*]] = load <8 x i64>, ptr [[TMP]]
; CHECK-NEXT: call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512(ptr
[[TMP2]], <8 x i64> [[TMP_VAL]])
; CHECK-NEXT: [[TMP4:%.*]] = load <8 x i64>, ptr [[TMP2]], align 32
; CHECK-NEXT: store <8 x i64> [[TMP4]], ptr [[ARG]], align 2
; CHECK-NEXT: ret void
;
bb:
    %tmp = alloca <8 x i64>, align 32
    %tmp2 = alloca <8 x i64>, align 32
    call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
    call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512(ptr %tmp2, ptr %tmp)
    %tmp4 = load <8 x i64>, ptr %tmp2, align 32
    store <8 x i64> %tmp4, ptr %arg, align 2
    ret void
}

; This should promote
define
    internal fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256(ptr %arg, ptr readonly
%arg1) #1 {
; CHECK-LABEL: define {{{^@}+}}@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]], <8 x i64> [[ARG1_VAL:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: store <8 x i64> [[ARG1_VAL]], ptr [[ARG]]
; CHECK-NEXT: ret void
;
bb:
    %tmp = load <8 x i64>, ptr %arg1
    store <8 x i64> %tmp, ptr %arg
    ret void
}

```

```

define void @avx512_legal512_prefer256_call_avx512_legal512_prefer256(ptr %arg) #1 {
; CHECK-LABEL: define {{{^@}+}}@avx512_legal512_prefer256_call_avx512_legal512_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: [[TMP2:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: call void @llvm.memset.p0.i64(ptr align 32 [[TMP]], i8 0, i64 32, i1 false)
; CHECK-NEXT: [[TMP_VAL:%.*]] = load <8 x i64>, ptr [[TMP]]
; CHECK-NEXT: call fastcc
void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256(ptr [[TMP2]], <8 x i64>
[[TMP_VAL]])
; CHECK-NEXT: [[TMP4:%.*]] = load <8 x i64>, ptr [[TMP2]], align 32
; CHECK-NEXT: store <8 x i64> [[TMP4]], ptr [[ARG]], align 2
; CHECK-NEXT: ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align 32
call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256(ptr %tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

; This should promote
define internal fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256(ptr %arg, ptr
readonly %arg1) #1 {
; CHECK-LABEL: define {{{^@}+}}@callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]], <8 x i64> [[ARG1_VAL:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: store <8 x i64> [[ARG1_VAL]],
ptr [[ARG]]
; CHECK-NEXT: ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

define void @avx512_legal512_prefer512_call_avx512_legal512_prefer256(ptr %arg) #0 {
; CHECK-LABEL: define {{{^@}+}}@avx512_legal512_prefer512_call_avx512_legal512_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: [[TMP2:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: call void @llvm.memset.p0.i64(ptr align 32 [[TMP]], i8 0, i64 32, i1 false)

```



```

; CHECK-NEXT: [[TMP_VAL:%.*]] = load <8 x i64>, ptr [[TMP]]
; CHECK-NEXT: call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256(ptr
[[TMP2]], <8 x i64> [[TMP_VAL]])
; CHECK-NEXT: [[TMP4:%.*]] = load <8 x i64>, ptr [[TMP2]], align 32
; CHECK-NEXT: store <8 x i64> [[TMP4]], ptr [[ARG]], align 2
; CHECK-NEXT: ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align
32
call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256(ptr %tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

```

; This should promote

```

define internal fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512(ptr %arg, ptr
readonly %arg1) #0 {
; CHECK-LABEL: define {{^[^@]+}}@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512
; CHECK-SAME: (ptr [[ARG:%.*]], <8 x i64> [[ARG1_VAL:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: store <8 x i64> [[ARG1_VAL]], ptr [[ARG]]
; CHECK-NEXT: ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

```

```

define void @avx512_legal512_prefer256_call_avx512_legal512_prefer512(ptr %arg) #1 {
; CHECK-LABEL: define {{^[^@]+}}@avx512_legal512_prefer256_call_avx512_legal512_prefer512
; CHECK-SAME: (ptr [[ARG:%.*]])
; CHECK-NEXT:
bb:
; CHECK-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: [[TMP2:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: call void @llvm.memset.p0.i64(ptr align 32 [[TMP]], i8 0, i64 32, i1 false)
; CHECK-NEXT: [[TMP_VAL:%.*]] = load <8 x i64>, ptr [[TMP]]
; CHECK-NEXT: call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512(ptr
[[TMP2]], <8 x i64> [[TMP_VAL]])
; CHECK-NEXT: [[TMP4:%.*]] = load <8 x i64>, ptr [[TMP2]], align 32
; CHECK-NEXT: store <8 x i64> [[TMP4]], ptr [[ARG]], align 2
; CHECK-NEXT: ret void
;

```

```

bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align 32
call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512(ptr %tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

; This should not promote
define internal fastcc void @callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr
%arg, ptr readonly %arg1) #1 {
; CHECK-LABEL: define {{^[^@]+}}@callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]], ptr readonly [[ARG1:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: [[TMP:%.*]] = load <8 x i64>, ptr [[ARG1]]
; CHECK-NEXT: store <8 x i64> [[TMP]], ptr [[ARG]]
; CHECK-NEXT: ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

define void @avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr %arg) #2 {
; CHECK-LABEL: define {{^[^@]+}}@avx512_legal256_prefer256_call_avx512_legal512_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: [[TMP2:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: call void @llvm.memset.p0.i64(ptr align 32 [[TMP]], i8 0, i64 32, i1 false)
; CHECK-NEXT: call fastcc void @callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr
[[TMP2]], ptr [[TMP]])
; CHECK-NEXT: [[TMP4:%.*]] = load <8 x i64>, ptr [[TMP2]], align 32
; CHECK-NEXT: store <8 x i64> [[TMP4]], ptr [[ARG]], align 2
; CHECK-NEXT: ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align 32
call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
call fastcc void @callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr %tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

```

```

; This should not promote
define internal fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr %arg, ptr
readonly %arg1) #2 {
; CHECK-LABEL: define {{^[^@]+}}@callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]], ptr readonly [[ARG1:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: [[TMP:%.*]] = load <8 x i64>, ptr [[ARG1]]
; CHECK-NEXT: store <8 x i64>
[[TMP]], ptr [[ARG]]
; CHECK-NEXT: ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

define void @avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr %arg) #1 {
; CHECK-LABEL: define {{^[^@]+}}@avx512_legal512_prefer256_call_avx512_legal256_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: [[TMP2:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: call void @llvm.memset.p0.i64(ptr align 32 [[TMP]], i8 0, i64 32, i1 false)
; CHECK-NEXT: call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr
[[TMP2]], ptr [[TMP]])
; CHECK-NEXT: [[TMP4:%.*]] = load <8 x i64>, ptr [[TMP2]], align 32
; CHECK-NEXT: store <8 x i64> [[TMP4]], ptr [[ARG]], align 2
; CHECK-NEXT: ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align 32
call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64
32, i1 false)
call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr %tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

; This should promote
define internal fastcc void @callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256(ptr %arg, ptr readonly
%arg1) #3 {
; CHECK-LABEL: define {{^[^@]+}}@callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]], <8 x i64> [[ARG1_VAL:%.*]])
; CHECK-NEXT: bb:

```

```

; CHECK-NEXT: store <8 x i64> [[ARG1_VAL]], ptr [[ARG]]
; CHECK-NEXT: ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void
}

define void @avx2_legal256_prefer256_call_avx2_legal512_prefer256(ptr %arg) #4 {
; CHECK-LABEL: define {{^[^@]+}}@avx2_legal256_prefer256_call_avx2_legal512_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT:
[[TMP2:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: call void @llvm.memset.p0.i64(ptr align 32 [[TMP]], i8 0, i64 32, i1 false)
; CHECK-NEXT: [[TMP_VAL:%.*]] = load <8 x i64>, ptr [[TMP]]
; CHECK-NEXT: call fastcc void @callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256(ptr
[[TMP2]], <8 x i64> [[TMP_VAL]])
; CHECK-NEXT: [[TMP4:%.*]] = load <8 x i64>, ptr [[TMP2]], align 32
; CHECK-NEXT: store <8 x i64> [[TMP4]], ptr [[ARG]], align 2
; CHECK-NEXT: ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align 32
call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
call fastcc void @callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256(ptr %tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

; This should promote
define internal fastcc void @callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256(ptr %arg, ptr readonly
%arg1) #4 {
;
CHECK-LABEL: define {{^[^@]+}}@callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]], <8 x i64> [[ARG1_VAL:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: store <8 x i64> [[ARG1_VAL]], ptr [[ARG]]
; CHECK-NEXT: ret void
;
bb:
%tmp = load <8 x i64>, ptr %arg1
store <8 x i64> %tmp, ptr %arg
ret void

```

```

}

define void @avx2_legal512_prefer256_call_avx2_legal256_prefer256(ptr %arg) #3 {
; CHECK-LABEL: define {{^@}+}}@avx2_legal512_prefer256_call_avx2_legal256_prefer256
; CHECK-SAME: (ptr [[ARG:%.*]])
; CHECK-NEXT: bb:
; CHECK-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: [[TMP2:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT: call void @llvm.memset.p0.i64(ptr align 32 [[TMP]], i8 0, i64 32, i1 false)
; CHECK-NEXT: [[TMP_VAL:%.*]] = load <8 x i64>, ptr [[TMP]]
; CHECK-NEXT: call fastcc void @callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256(ptr
[[TMP2]], <8 x i64> [[TMP_VAL]])
; CHECK-NEXT: [[TMP4:%.*]] = load
<8 x i64>, ptr [[TMP2]], align 32
; CHECK-NEXT: store <8 x i64> [[TMP4]], ptr [[ARG]], align 2
; CHECK-NEXT: ret void
;
bb:
%tmp = alloca <8 x i64>, align 32
%tmp2 = alloca <8 x i64>, align 32
call void @llvm.memset.p0.i64(ptr align 32 %tmp, i8 0, i64 32, i1 false)
call fastcc void @callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256(ptr %tmp2, ptr %tmp)
%tmp4 = load <8 x i64>, ptr %tmp2, align 32
store <8 x i64> %tmp4, ptr %arg, align 2
ret void
}

; If the arguments are scalar, its ok to promote.
define internal i32 @scalar_callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr %X, ptr %Y)
#2 {
; CHECK-LABEL: define
{{^@}+}}@scalar_callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256
; CHECK-SAME: (i32 [[X_VAL:%.*]], i32 [[Y_VAL:%.*]])
; CHECK-NEXT: [[C:%.*]] = add i32 [[X_VAL]], [[Y_VAL]]
; CHECK-NEXT: ret i32 [[C]]
;
%A = load i32, ptr %X
%B = load i32, ptr %Y
%C = add i32 %A, %B
ret i32 %C
}

define i32 @scalar_avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr
%B) #2 {
; CHECK-LABEL: define {{^@}+}}@scalar_avx512_legal256_prefer256_call_avx512_legal512_prefer256
; CHECK-SAME: (ptr [[B:%.*]])
; CHECK-NEXT: [[A:%.*]] = alloca i32
; CHECK-NEXT: store i32 1, ptr [[A]]

```

```

; CHECK-NEXT: [[A_VAL:%.*]] = load i32, ptr [[A]]
; CHECK-NEXT: [[B_VAL:%.*]] = load i32, ptr [[B]]
; CHECK-NEXT: [[C:%.*]] = call i32
@scalar_callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(i32 [[A_VAL]], i32 [[B_VAL]])
; CHECK-NEXT: ret i32 [[C]]
;
%A = alloca i32
store i32 1, ptr %A
%C = call i32 @scalar_callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(ptr %A, ptr %B)
ret i32 %C
}

; If the arguments are scalar, its ok to promote.
define internal i32 @scalar_callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr %X, ptr %Y)
#2 {
; CHECK-LABEL: define
{{{^@}+}}@scalar_callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256
; CHECK-SAME: (i32 [[X_VAL:%.*]],
i32 [[Y_VAL:%.*]])
; CHECK-NEXT: [[C:%.*]] = add i32 [[X_VAL]], [[Y_VAL]]
; CHECK-NEXT: ret i32 [[C]]
;
%A = load i32, ptr %X
%B = load i32, ptr %Y
%C = add i32 %A, %B
ret i32 %C
}

define i32 @scalar_avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr %B) #2 {
; CHECK-LABEL: define {{{^@}+}}@scalar_avx512_legal512_prefer256_call_avx512_legal256_prefer256
; CHECK-SAME: (ptr [[B:%.*]])
; CHECK-NEXT: [[A:%.*]] = alloca i32
; CHECK-NEXT: store i32 1, ptr [[A]]
; CHECK-NEXT: [[A_VAL:%.*]] = load i32, ptr [[A]]
; CHECK-NEXT: [[B_VAL:%.*]] = load i32, ptr [[B]]
; CHECK-NEXT: [[C:%.*]] = call i32
@scalar_callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(i32 [[A_VAL]], i32 [[B_VAL]])
; CHECK-NEXT: ret i32 [[C]]
;
%A = alloca i32
store i32 1, ptr %A
%C = call i32 @scalar_callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(ptr %A, ptr %B)
ret i32 %C
}

; Function Attrs: argmemonly nounwind
declare void @llvm.memset.p0.i64(ptr

```

nocapture writeonly, i8, i64, i1) #5

```
attributes #0 = { inlinehint norecurse nounwind uwtable "target-features"="+avx512vl" "min-legal-vector-width"="512" "prefer-vector-width"="512" }
attributes #1 = { inlinehint norecurse nounwind uwtable "target-features"="+avx512vl" "min-legal-vector-width"="512" "prefer-vector-width"="256" }
attributes #2 = { inlinehint norecurse nounwind uwtable "target-features"="+avx512vl" "min-legal-vector-width"="256" "prefer-vector-width"="256" }
attributes #3 = { inlinehint norecurse nounwind uwtable "target-features"="+avx2" "min-legal-vector-width"="512" "prefer-vector-width"="256" }
attributes #4 = { inlinehint norecurse nounwind uwtable "target-features"="+avx2" "min-legal-vector-width"="256" "prefer-vector-width"="256" }
attributes #5 = { argmemonly nounwind }
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=riscv64 -mattr=+zbc -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64ZBC-ZBKC
; RUN: llc -mtriple=riscv64 -mattr=+zbbc -verify-machineinstrs < %s \
; RUN: -riscv-experimental-rv64-legal-i32 | FileCheck %s -check-prefix=RV64ZBC-ZBKC
```

declare i64 @llvm.riscv.clmul.i64(i64 %a, i64 %b)

```
define i64 @clmul64(i64 %a, i64 %b) nounwind {
; RV64ZBC-ZBKC-LABEL: clmul64:
; RV64ZBC-ZBKC:      # %bb.0:
; RV64ZBC-ZBKC-NEXT:  clmul a0, a0, a1
; RV64ZBC-ZBKC-NEXT:  ret
  %tmp = call i64 @llvm.riscv.clmul.i64(i64 %a, i64 %b)
  ret i64 %tmp
}
```

declare i64 @llvm.riscv.clmulh.i64(i64 %a, i64 %b)

```
define i64 @clmul64h(i64 %a, i64 %b) nounwind {
; RV64ZBC-ZBKC-LABEL: clmul64h:
; RV64ZBC-ZBKC:      # %bb.0:
; RV64ZBC-ZBKC-NEXT:  clmulh a0, a0, a1
; RV64ZBC-ZBKC-NEXT:  ret
  %tmp = call i64 @llvm.riscv.clmulh.i64(i64 %a, i64 %b)
  ret i64 %tmp
}
```

declare

i32 @llvm.riscv.clmul.i32(i32 %a, i32 %b)

```
define signext i32 @clmul32(i32 signext %a, i32 signext %b) nounwind {
; RV64ZBC-ZBKC-LABEL: clmul32:
; RV64ZBC-ZBKC:      # %bb.0:
; RV64ZBC-ZBKC-NEXT:  clmul a0, a0, a1
```

```

; RV64ZBC-ZBKC-NEXT:  sext.w a0, a0
; RV64ZBC-ZBKC-NEXT:  ret
    %tmp = call i32 @llvm.riscv.clmul.i32(i32 %a, i32 %b)
    ret i32 %tmp
}

declare i32 @llvm.riscv.clmulh.i32(i32 %a, i32 %b)

define signext i32 @clmul32h(i32 signext %a, i32 signext %b) nounwind {
; RV64ZBC-ZBKC-LABEL: clmul32h:
; RV64ZBC-ZBKC:      # %bb.0:
; RV64ZBC-ZBKC-NEXT:  slli a1, a1, 32
; RV64ZBC-ZBKC-NEXT:  slli a0, a0, 32
; RV64ZBC-ZBKC-NEXT:  clmulh a0, a0, a1
; RV64ZBC-ZBKC-NEXT:  srai a0, a0, 32
; RV64ZBC-ZBKC-NEXT:  ret
    %tmp = call i32 @llvm.riscv.clmulh.i32(i32 %a, i32 %b)
    ret i32 %tmp
}

; FIXME: We could avoid the slli instructions by using clmul+srai since the
; inputs are zero extended.
define signext i32 @clmul32h_zext(i32 zeroext %a, i32 zeroext %b) nounwind {
;
; RV64ZBC-ZBKC-LABEL: clmul32h_zext:
; RV64ZBC-ZBKC:      # %bb.0:
; RV64ZBC-ZBKC-NEXT:  slli a1, a1, 32
; RV64ZBC-ZBKC-NEXT:  slli a0, a0, 32
; RV64ZBC-ZBKC-NEXT:  clmulh a0, a0, a1
; RV64ZBC-ZBKC-NEXT:  srai a0, a0, 32
; RV64ZBC-ZBKC-NEXT:  ret
    %tmp = call i32 @llvm.riscv.clmulh.i32(i32 %a, i32 %b)
    ret i32 %tmp
}

```

# 1.186 jq 1.8.1-r0

## 1.186.1 Available under license :

Oniguruma LICENSE

-----

Copyright (c) 2002-2021 K.Kosako <kkosako0@gmail.com>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:



1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

jq is copyright (C) 2012 Stephen Dolan

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

jq's documentation (everything found under the docs/ subdirectory in the source tree) is licensed under the Creative Commons CC BY 3.0 license, which can be found at:

<https://creativecommons.org/licenses/by/3.0/>

The documentation website includes a copy of Twitter's Bootstrap and relies on Bonsai, Liquid templates and various other projects, look them up for detailed licensing conditions.

jq incorporates David M. Gay's `dtoa.c` and `g_fmt.c`, which bear the following notices:

`dtoa.c`:

The author of this software is David M. Gay.

Copyright (c) 1991, 2000, 2001 by Lucent Technologies.

Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software.

THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHOR NOR LUCENT MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

`g_fmt.c`:

The author of this software is David M. Gay.

Copyright (c) 1991, 1996 by Lucent Technologies.

Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software.

THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHOR NOR LUCENT MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

jq uses parts of the open source C library "decNumber", which is distributed under the following license:

ICU License - ICU 1.8.1 and later

## COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2005 International Business Machines Corporation and others  
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

-----  
All trademarks and registered trademarks mentioned herein are the property of their respective owners.

jv\_thread.h is copied from Heimdal's lib/base/heimbase.h and some code in jv.c is copied from Heimdal's lib/base/dll.c:

Portions Copyright (c) 2016 Kungliga Tekniska  
Hogskolan  
(Royal Institute of Technology, Stockholm, Sweden).  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

jq uses a modified version of NetBSD implementation strptime(), which is distributed under the following license:

Copyright (c) 1997, 1998, 2005, 2008 The NetBSD Foundation, Inc.  
All rights reserved.

This code was contributed to The NetBSD Foundation by Klaus Klein.  
Heavily optimised by David Laight

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED

TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.187 busybox 1.37.0-r20

### 1.187.1 Available under license :

bzip2 applet in busybox is based on lightly-modified source of bzip2 version 1.0.4. bzip2 source is distributed under the following conditions (copied verbatim from LICENSE file)

=====

This program, "bzip2", the associated library "libbzip2", and all documentation, are copyright (C) 1996-2006 Julian R Seward. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY

DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, Cambridge, UK.

[jseward@bzip.org](mailto:jseward@bzip.org)

bzip2/libbzip2 version 1.0.4 of 20 December 2006

--- A note on GPL versions

BusyBox is distributed under version 2 of the General Public License (included in its entirety, below). Version 2 is the only version of this license which this version of BusyBox (or modified versions derived from this one) may be distributed under.

---

## GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's

software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid

anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this

License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based



on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under

any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity

of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make

exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

## NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of

MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of  
Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

# 1.188 bind 9.20.16-r0

## 1.188.1 Available under license :

Mozilla Public License, version 2.0

### 1. Definitions

1.1. "Contributor"

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. "Contributor Version"

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"

means Covered Software of a particular Contributor.

1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"

means

a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. "Executable Form"

means any form of the work other than Source Code Form.

1.7. "Larger Work"

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. "License"

means this document.

1.9. "Licensable"

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

#### 1.10. "Modifications"

means any of the following:

- a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or
- b. any new file in Source Code Form that contains any Covered Software.

#### 1.11. "Patent Claims" of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims,

in any patent Licensable by such

Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

#### 1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

#### 1.13. "Source Code Form"

means the form of the work preferred for making modifications.

#### 1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by

contract or

otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

## 2. License Grants and Conditions

### 2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- a. under intellectual property rights (other than patent or trademark)  
Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

## 2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

## 2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or
- b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

## 2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if



permitted under the terms of Section 3.3).

## 2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

## 2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

## 2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

## 3. Responsibilities

### 3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

### 3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license

for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

### 3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

### 3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

### 3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

## 4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

## 5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly

and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

## 6. Disclaimer of Warranty

Covered Software is provided under this License on an "as is" basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

## 7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including

negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

## 8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

## 9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

## 10. Versions of the License

### 10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section

10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

### 10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license

steward.

### 10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

### 10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses If You choose to distribute Source Code Form that is Incompatible

With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

#### Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v.

2.0. If a copy of the MPL was not distributed with this file, You can obtain one at

<https://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

#### Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

Copyright (C) 1996-2023 Internet Systems Consortium, Inc. ("ISC")

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, you can obtain one at <https://mozilla.org/MPL/2.0/>.

-----

Portions of this code release fall under one or more of the following Copyright notices. Please see individual source

files for details.

For binary releases also see: OpenSSL-LICENSE.

Copyright (C) 1996-2001 Nominum, Inc.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND NOMINUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL NOMINUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

-----  
Copyright (C) 1995-2000 by Network Associates, Inc.

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND ISC AND NETWORK ASSOCIATES DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL ISC BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

-----  
Copyright (C) 2002 Stichting NLnet, Netherlands, stichting@nlnet.nl.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND STICHTING NLNET DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL

STICHTING NLNET BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The development of Dynamically Loadable Zones (DLZ) for Bind 9 was conceived and contributed by Rob Butler.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND ROB BUTLER DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL ROB BUTLER BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

-----  
Copyright (c) 1987, 1990, 1993, 1994

The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES

(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-----

Copyright (C) The Internet Society 2005. This version of this module is part of RFC 4178; see the RFC itself for full legal notices.

(The above copyright notice is per RFC 3978 5.6 (a), q.v.)

-----

Copyright (c) 2004 Masarykova universita  
(Masaryk University, Brno, Czech Republic)  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.



-----  
Copyright (c) 1997 - 2003 Kungliga Tekniska Hgskolan  
(Royal Institute of Technology, Stockholm, Sweden).  
All rights reserved.

Redistribution and use in source and binary forms, with or without  
modification, are permitted provided that the following conditions  
are met:

1. Redistributions of source code must retain the above copyright  
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright  
notice, this list of conditions and the following disclaimer in the  
documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute  
nor the names of its contributors  
may be used to endorse or promote products derived from this software  
without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS ``AS IS" AND  
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE  
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL  
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS  
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT  
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY  
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF  
SUCH DAMAGE.

-----  
Copyright (c) 1993  
by Digital Equipment Corporation.

Permission to use, copy, modify, and distribute this software for any  
purpose with or without fee is hereby granted, provided that the above  
copyright notice and this permission notice appear in all copies, and that  
the name of Digital Equipment Corporation not be used in advertising or  
publicity pertaining to distribution of the document or software without  
specific, written prior permission.

THE SOFTWARE IS PROVIDED "AS IS" AND DIGITAL EQUIPMENT CORP. DISCLAIMS ALL

WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL DIGITAL EQUIPMENT CORPORATION BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

-----  
Copyright

(C) 1995, 1996, 1997, and 1998 WIDE Project.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the project nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE PROJECT AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

DISCLAIMED. IN NO EVENT SHALL THE PROJECT OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-----  
Copyright (c) 1999-2000 by Nortel Networks Corporation

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND NORTEL NETWORKS DISCLAIMS

ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL NORTEL NETWORKS BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

-----

Copyright (C) 2004 Nominet, Ltd.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND NOMINET DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL ISC BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

-----

Copyright (c) 1996, David Mazieres <dm@uun.org>  
Copyright (c) 2008, Damien Miller <djm@openbsd.org>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

-----

Copyright (c) 1995, 1997, 1998 The NetBSD Foundation, Inc.  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS BE LIABLE

FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-----  
Copyright (C) 2008-2011 Red Hat, Inc.

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND Red Hat DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL Red Hat BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

-----  
Copyright (c) 2013-2014, Farsight Security, Inc.  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-----  
Copyright (c) 2014 by Farsight Security, Inc.

Licensed under the Apache License,  
Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

-----  
Copyright Joyent, Inc. and other Node contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to

deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

©2026 Cisco Systems, Inc. All rights reserved.